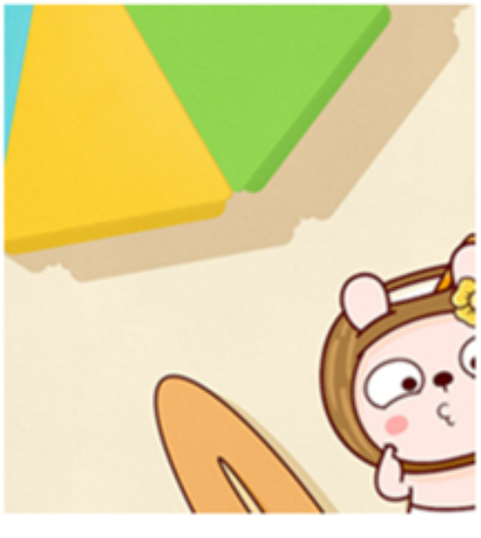
最新IT书籍PDF
最新IT书籍PDF
最新IT书籍PDF
最新IT书籍PDF



版权相关注意事项：
1、书籍版权归著者和出版社所有
2、本PDF来自于各个广泛的信息平台，经过整理而成
3、本PDF仅限用于非商业用途或者个人交流研究学习使用
4、本PDF获得者不得在互联网上以任何目的进行传播，违规者造成的法律责任和后果，违规者自负
5、如果觉得书籍内容很赞，请一定购买正版实体书，多多支持编写高质量的图书的作者和相应的出版社！当然，如果图书内容不堪入目，质量低下，你也可以选择狠狠滴撕裂本PDF
6、技术类书籍是拿来获取知识的，不是拿来收藏的，你得到了书籍不意味着你得到了知识，所以请不要得到书籍后就觉得沾沾自喜，要经常翻阅！！经常翻阅
7、请于下载PDF后24小时内研究使用并删掉本PDF

版权相关注意事项：
1、书籍版权归著者和出版社所有
2、本PDF来自于各个广泛的信息平台，经过整理而成
3、本PDF仅限用于非商业用途或者个人交流研究学习使用
4、本PDF获得者不得在互联网上以任何目的进行传播，违规者造成的法律责任和后果，违规者自负
5、如果觉得书籍内容很赞，请一定购买正版实体书，多多支持编写高质量的图书的作者和相应的出版社！当然，如果图书内容不堪入目，质量低下，你也可以选择狠狠滴撕裂本PDF
6、技术类书籍是拿来获取知识的，不是拿来收藏的，你得到了书籍不意味着你得到了知识，所以请不要得到书籍后就觉得沾沾自喜，要经常翻阅！！经常翻阅
7、请于下载PDF后24小时内研究使用并删掉本PDF

仅供非商业用途或交流学习使用

HZ BOOKS 华章科技
CRC Press Taylor & Francis Group

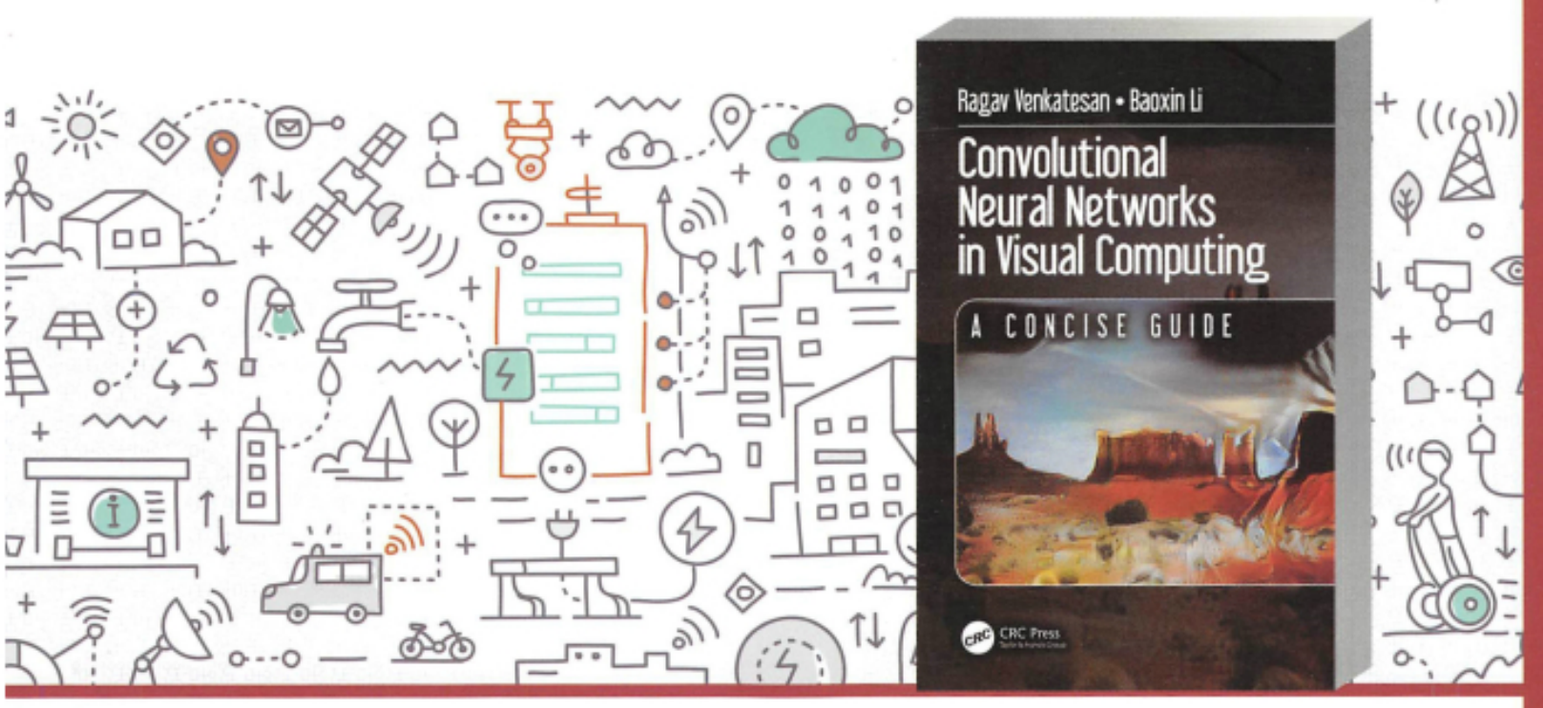
智能科学与技术丛书

Convolutional Neural Networks in Visual Computing

卷积神经网络与视觉计算

[美] 拉加夫·维凯特森（Ragav Venkatesan）
李宝新（Baoxin Li）◎著

钱亚冠 王滨 ◎等译



机械工业出版社
China Machine Press

仅供非商业用途或交流学习使用
仅供非商业用途或交流学习使用

作者简介

拉加夫·维凯特森（Ragav Venkatesan）

他是 ASU 的视觉表示和处理小组的一名研究助理，曾在机器学习、模式识别、视频处理和计算机视觉等多个研究生课程中担任助教。他是机器学习和计算机视觉领域的几个期刊的同行评审和会议审稿人。

李宝新（Baoxin Li）

目前是亚利桑那州立大学计算机科学与工程专业的教授和主任。从 2000 年到 2004 年，他任美国夏普实验室的高级研究员，领导研发夏普的 HiMPACT 体育技术。他还曾是俄勒冈州波特兰州立大学的副教授。他拥有 18 项美国专利，目前的研究兴趣包括计算机视觉和模式识别、多媒体、社会计算、机器学习和辅助技术。2001 年和 2004 年，他获得夏普实验室总裁奖。2002 年，他获得了夏普实验室年度发明者奖，还曾获得国家自然科学基金会职业奖。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

华章科技
HZBOOKS | Science & Technology

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Convolutional Neural Networks in Visual Computing

卷积神经网络与视觉计算

[美] 拉加夫·维凯特森（Ragav Venkatesan）
李宝新（Baoxin Li） ◎ 著
钱亚冠 王滨 ◎ 等译

机械工业出版社
China Machine Press

仅供非商业用途或交流学习使用

版权相关注意事项：
1、书籍版权归著者和出版社所有
2、本PDF来自于各个广泛的信息平台，经过整理而成
3、本PDF仅限用于非商业用途或者个人交流研究学习使用
4、本PDF获得者不得在互联网上以任何目的进行传播，违规者造成的法律责任和后果，违规者自负
5、如果觉得书籍内容很赞，请一定购买正版实体书，多多支持

编写高质量的图书的作者和相应的出版社！当然，如果图书内容不堪入目，质量低下，你也可以选择狠狠滴撕裂本PDF
6、技术类书籍是拿来获取知识的，不是拿来收藏的，你得到了书籍不意味着你得到了知识，所以请不要得到书籍后就觉得沾沾自喜，要经常翻阅！！经常翻阅
7、请于下载PDF后24小时内研究使用并删掉本PDF

仅供非商业用途或交流学习使用

图书在版编目（CIP）数据

卷积神经网络与视觉计算 /（美）拉加夫·维凯特森（Ragav Venkatesan），（美）李宝新著；钱亚冠等译 . —北京：机械工业出版社，2018.11
（智能科学与技术丛书）
书名原文：Convolutional Neural Networks in Visual Computing

ISBN 978-7-111-61239-1

I. 卷… II. ①拉… ②李… ③钱… III. 计算机视觉 - 人工神经网络 - 计算 IV. TP302.7

中国版本图书馆 CIP 数据核字（2018）第 250420 号

本书版权登记号：图字 01-2018-6729

Convolutional Neural Networks in Visual Computing by Ragav Venkatesan and Baoxin Li (9781138747951).
Copyright © 2018 by Taylor & Francis Group, LLC.

Authorized translation from the English language edition published by CRC Press, part of Taylor & Francis Group LLC. All rights reserved.

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版，并经授权翻译出版。版权所有，侵权必究。

本书中文简体字翻译版授权由机械工业出版社独家出版并仅限在中华人民共和国境内（不包括香港、澳门特别行政区及台湾地区）销售。未经出版者书面许可，不得以任何方式复制或抄袭本书的任何内容。

本书封面贴有 Taylor & Francis 公司防伪标签，无标签者不得销售。

卷积神经网络与视觉计算

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：陈佳媛　　责任校对：殷　虹
印　　刷：北京文昌阁彩色印刷有限责任公司　　版　　次：2019 年 1 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：11.5
书　　号：ISBN 978-7-111-61239-1　　定　　价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

版权所有 · 侵权必究
封底无防伪标均为盗版
本书法律顾问：北京大成律师事务所　韩光 / 邹晓东

仅供非商业用途或交流学习使用
仅供非商业用途或交流学习使用

感谢一同成长的 Jaikrishna Mohan，你是我忠实的朋友与兄弟；还有帮助我成长的 Ravi Naganathan 教授。你们的智慧哲学和人生信条使我更为出彩。

——Ragav Venkatesan

感谢我的爱人 Julie，感谢你对我长年坚定不移的支持。

——李宝新

仅供非商业用途或交流学习使用
仅供非商业用途或交流学习使用

译 者 序

Convolutional Neural Networks in Visual Computing

随着深度学习的兴起，卷积神经网络重新焕发出光彩，它被成功地应用于计算机视觉领域，并获得了前所未有的进步。不管是业界还是教育界，都希望有一本能快速学习这一技术的资料。但由于这一领域的发展很快，大多内容都散布在各种学术论文中，初学者要通读很多论文后才能掌握这些技术，这无疑是个令人痛苦的过程。在为实验室的研究生新生寻找一本合适的教材时，本人也同样感到万分苦恼。一个偶然的机会，我在亚马逊上发现了本书的英文版。本书从基本的图像知识讲起，到机器学习的核心理论，再到神经网络和卷积神经网络及在计算机视觉中的最新应用，可以说形成了一个非常完整的知识体系。尽管这些内容不管哪一部分单独都可以成为一本教材，但本书的优势就在于用浅显的语言把这些貌似深奥的内容，通过合理的裁剪，围绕视觉计算把逻辑上相关的内容整合到了一起，免去了初学者到处翻阅相关书籍的麻烦，也有助于读者在图像处理、机器学习、深度学习、视觉计算上形成完整的认知。

感谢机械工业出版社华章公司给予翻译这本前沿著作的机会。本书的翻译也得到了浙江省自然科学基金项目（No. LY17F020011）的资助。参加本书翻译的有浙江科技学院大数据学院的钱亚冠、杭州海康威视网络与信息安全实验室的王滨，以及人工智能与机器学习实验室的研究生：卢红波、马骏、马丹峰、郭艳凯、刘新伟、张锡敏等。由于本书内容较新，翻译时间仓促，难免有不足之处，恳请读者批评指正！联系邮箱 QianYaGuan@zust. edu. cn。

钱亚冠

2018 年 6 月于杭州

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Ragav Venkatesan

目前正在亚利桑那州立大学（ASU）的计算、信息学和决策系统工程学院攻读计算机科学博士学位。他是 ASU 的视觉表示和处理组的一名研究助理，曾担任机器学习、模式识别、视频处理和计算机视觉等多个研究生课程的助教。在此之前，他是 ASU 电子计算机工程学院图像处理与应用实验室的研究助理，2012 年获得了硕士学位。从 2013 年到 2014 年，他是英特尔公司的一名研究实习生，研究自动驾驶中的计算机视觉。他是机器学习和计算机视觉领域的几个期刊的同行评审和会议审稿人。

Baoxin Li

2000 年获马里兰大学帕克学院电气工程博士学位，目前是亚利桑那州立大学计算机科学与工程专业的教授和主任，以及电气工程和计算机工程专业的研究生导师。从 2000 年到 2004 年，他任美国夏普实验室的高级研究员，在那里他领导研发了夏普的 HiMPACT 体育技术。2003 年至 2004 年，他还是俄勒冈州波特兰州立大学的副教授。他拥有 18 项美国专利，目前的研究兴趣包括计算机视觉和模式识别、多媒体、社会计算、机器学习和辅助技术。2001 年和 2004 年，他获得夏普实验室总裁奖。2002 年，他获得了夏普实验室年度发明者奖，也是国家自然科学基金会职业奖的获得者。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

近年来，深度学习体系结构由于在计算机视觉等应用中的极大成功而开始流行起来。特别是卷积神经网络（CNN）已经成为深度学习中最重要的一种网络结构。现在普遍认为深度学习和 CNN 的术语对于初学者来说，太难学习和掌握了。为此，本书涵盖了计算机视觉中的深度学习、设计和部署 CNN，以及深度计算机视觉体系结构的基础知识。本书旨在为工程师、本科生和研究生提供入门指南，帮助他们快速学习和建立自己的深度学习系统。本书以通俗易懂的方式写成，目的是为计算机视觉相关的深度学习技术提供一个易学的教材，同时仍然涵盖了大量必要的基础知识。

本书分为 5 章。第 1 章简要介绍了图像表示和一些计算机视觉模型，这些模型现在被称为人工方式建模。本章为读者提供了对图像表示的基本理解，并介绍了一些线性和非线性的特征提取或表示方法，以及这些表示的特性。此外，本章还介绍了一些基本图像元素（如边缘）的检测方法，还包括用这些表示来完成一些基本的机器学习任务。本章最后对两种流行的非神经网络计算机视觉建模技术进行了介绍。

第 2 章介绍了回归、机器学习和优化的概念。本章首先介绍监督学习，介绍的第一个学习模型是线性回归，它的第一种求解方法是利用最小二乘法得到解析解，并用极大似然来进行解释。本章通过基函数展开的方式实现线性模型向非线性模型的转变，并进一步介绍了交叉验证和正则化的方法解决过拟合和泛化问题。本章的后半部分介绍了凸误差面和非凸误差面的梯度下降最优化方法，进一步介绍了各种梯度下降法和正则化几何，并对基本梯度下降法进行了改进，包括二阶最小化损失技术和动量学习。

仅供非商业用途或交流学习使用

版权相关注意事项：
1、书籍版权归著者和出版社所有
2、本PDF来自于各个广泛的信息平台，经过整理而成
3、本PDF仅限用于非商业用途或者个人交流研究学习使用
4、本PDF获得者不得在互联网上以任何目的进行传播，违规者造成的法律责任和后果，违规者自负
5、如果觉得书籍内容很赞，请一定购买正版实体书，多多支持编写高质量的图书的作者和相应的出版社！当然，如果图书内容不堪入目，质量低下，你也可以选择狠狠滴撕裂本PDF
6、技术类书籍是拿来获取知识的，不是拿来收藏的，你得到了书籍不意味着你得到了知识，所以请不要得到书籍后就觉得沾沾自喜，要经常翻阅！！经常翻阅
7、请于下载PDF后24小时内研究使用并删掉本PDF

仅供非商业用途或交流学习使用

第3章和第4章是本书的核心。第3章在第2章的基础上，介绍了Rosenblatt感知器和感知器学习算法。本章接着介绍了logistic神经元及其激活函数，研究了两类和多类问题的单神经元模型，以及这种神经元的优缺点，并介绍了异或问题。针对异或问题，提出了多层神经网络的思想，介绍了反向传播算法，并对其进行了改进，给出了一些实用技巧，有助于工程上更好、更稳定的实现。第4章介绍了卷积池化层和CNN。其中研究了该层的各种属性，并分析了典型的数字手写体数据集提取的特征。本章还介绍了目前最流行的四种CNN：AlexeNet、VGG、GoogLeNet和ResNet，并对它们的架构和思想进行了比较。

第5章通过介绍CNN的一些当前新颖实用的用法，进一步拓展和丰富了深度神经网络的结构。本章大致分为两个连续的部分。第一部分讨论了使用网上可下载，并已预训练好的现成深度网络的原理。经过预训练的网络是在一个完整的数据集上进行训练得到的，公开提供给研究人员用于新的机器学习任务。这些都是在满足通用、可迁移的条件下研究的。第5章还研究了给定一个预训练网络的情况下，压缩网络的和学习新任务的方法。第二部分论述了CNN不用于监督学习，而是用于生成网络的思想，简要研究了自编码器和最新的计算机视觉技术：生成对抗网络（GAN）。

本书的相关网站（convolution.network）包含代码及实现（也可登录www.hzbook.com下载）、部分彩色插图、勘误表和补充资料。2017年春季，本书作为亚利桑那州立大学的一门研究生课程的教材进行讲授，讲座和材料也可以在本书网站上找到。

图1-1是我拍摄的原始图像（original.jpg）。这是monument山谷的一幅图片，它是美国西南地区的代表，亚利桑那州也在那里。memory.png是以Salvador Dali的风格绘制的，特别是他的绘画《记忆的持久性》，它抽象地描述了心灵幻觉的概念，以无形的形式描绘和处理物体，很像我们在书中研究的神经网络的一些表现

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

形式。

memory. png 不是人类所画，而是由本书介绍的神经网络绘制的。如果引用了本书，请使用以下的引文参考方式。

```
@article{DBLP:journals/corr/GatysEB15a,
  author    = {Leon A. Gatys and
               Alexander S. Ecker and
               Matthias Bethge},
  title     = {A Neural Algorithm of Artistic Style},
  journal   = {CoRR},
  volume    = {abs/1508.06576},
  year      = {2015},
  url       = {http://arxiv.org/abs/1508.06576},
  timestamp = {Wed, 07 Jun 2017 14:41:58 +0200},
  biburl    = {http://dblp.unitrier.de/rec/bib/
               journals/corr/GatysEB15a},
  bibsource = {dblp computer science bibliography,
               http://dblp.org}
}
```

本书配有一个基于 Python 和 Theano 的 CNN 工具箱（该工具箱是由本书作者开发的），还有一个网页包含彩色图片、勘误表和其他补充资料。这个神经网络工具箱被命名为 yann，可以在 MIT 许可证下使用，网址为 http://www. yann. network。为了让初学者更容易地阅读书中的内容，作者开发了一套使用 yann 的教程。该教程和工具箱涵盖了本书讨论的不同体系结构的计算机，并提供了示例代码和应用程序编程接口（API）文档。在撰写本书的时候，yann 工具箱正在积极开发中，它的客户支持是通过 GitHub 提供的。本书的网页在 http://guide2cnn. com 上。书中的大多数图片都是灰度图，但有些图片最初是彩色的，是在制作过程中被转换成了灰度图。这些图片的彩色版本以及附加说明、相关课程的信息以及问答也可在网站上找到。

工具箱和本书将作为亚利桑那州立大学一学期的“深入学习视觉计算”研究生课程的阅读材料。本课程包括录制的讲座、课程内容和作业，可供大家使用，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

网址为 http://www. course. convolution. network. 。作者可以通过电子邮件回答大家对有关内容和代码的问题，虚心接受大家对本书内容的批评或评论。只要符合出版商的版权政策，作者允许复制书中的图片、结果和内容。作者希望读者喜

欢这本书，并希望初学者能够在本书及工具箱的帮助下快速构建自己的学习机器。鼓励读者利用本书中的知识做有益于人类的事，但同时真诚地劝告他们不要建造“Skynet”或任何其他末日人工智能机器。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

感谢许多同事，是他们使这个耗时的写书项目的完成变得可能并使我从中获得快乐。亚利桑那州立大学的视觉表示与处理组和认知与普适计算中心的许多当前和过去的研究成员都致力于深度学习的各个方面及其在视觉计算方面的应用。他们的努力为本书的写作提供了有见解的内容，对此我们表示非常感激。特别要感谢 Parag Sridhar Chandakkar 对第 4 章和第 5 章提出的建议，以及 Yuzhen Ding、Yikang Li、Vijetha Gattupalli 和 Hemanth Venkateswara，他们总能提供极为有益的观点。

这项工作源于美国国家自然科学基金、海军研究办公室、陆军研究办公室和诺基亚赞助的几个项目。本书的任何观点或结论仅代表作者本人，与赞助者无关。我们也感谢 NVIDIA 公司的支持，他们捐赠的 Tesla K40 GPU 为我们的研究工作提供了帮助。

感谢 Taylor & Francis 出版集团、CRC 出版公司，特别是执行编辑 Cindy Carelli 和 Renee Nakash，感谢他们在本书的写作过程中所给予的耐心和极大支持。我们还要感谢 Alex Krizhevsky 博士，他大度地允许我们使用 AlexNet 论文中的数据。我们还要进一步感谢 Theano 和其他 Python 库的开发人员，这些库被 yann 工具箱所使用，并用于制作本书中的一些图片。特别地，我们要感谢来自 Theano 用户小组的 Frédéric Bastien 和 Pascal Lamblin，以及蒙特利尔大学机器学习算法学院的支持。我们还想感谢 GitHub、Read the Docs 为我们提供免费的数据、代码、文档和教程的在线托管。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

最后，也是最重要的，感谢我们的朋友和家人在这个项目中所给予的坚定支持，以及他们对我们占用许多周末和漫漫长夜的理解和宽容。我们把爱和这本书献给他们。

Ragav Venkatesan 和 Baoxin Li

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

目　录

Convolutional Neural Networks in Visual Computing

译者序

作者简介

前言

致谢

第 1 章　视觉计算简介 …… 1

1.1　图像表示基础 …… 3

1.1.1　变换域表示 …… 6

1.1.2　图像的直方图 …… 7

1.1.3　图像梯度和边缘 …… 10

1.1.4　超越图像梯度 …… 15

1.2　基于 Hough 变换的直线检测 …… 16

1.3　Harris 角点 …… 16

1.4　尺度不变的特征变换 …… 17

1.5　方向梯度直方图 …… 18

1.5.1　人工设计特征空间中的决策制定 …… 19

1.5.2　贝叶斯决策 …… 21

1.5.3　线性决策边界 …… 23

1.6　可变形零件模型的实例研究 …… 25

1.7　计算机视觉向神经网络转变 …… 27

本章小结 …… 30

参考文献 …… 30

第 2 章　回归问题中的机器学习 …… 34

2.1　监督学习 …… 34

2.2　线性模型 …… 38

2.3　最小二乘法 …… 40

2.4　极大似然估计的解释 …… 43

2.5　扩展到非线性模型 …… 45

2.6　正则化 …… 47

2.7　交叉验证 …… 50

2.8　梯度下降 …… 52

2.9　几何正则化 …… 58

2.10　非凸误差面 …… 60

2.11　随机梯度、批梯度及在线梯度下降 …… 61

2.12　其他自适应学习率的更新规则 …… 62

2.13　动量 …… 64

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

本章小结 …… 65
参考文献 …… 66

第 3 章 人工神经网络 …… 69
3.1 感知器 …… 70
3.2 多层神经网络 …… 78
3.3 反向传播算法 …… 82
3.4 改进的反向传播算法 …… 85
3.4.1 激活函数 …… 86
3.4.2 权重剪枝 …… 88
3.4.3 批量标准化 …… 89
本章小结 …… 90
参考文献 …… 90

第 4 章 卷积神经网络 …… 93
4.1 卷积与池化层 …… 94
4.2 卷积神经网络 …… 102
本章小结 …… 118
参考文献 …… 119

第 5 章 卷积神经网络的新进展 …… 121
5.1 预训练网络 …… 122
5.1.1 通用性和可传递性 …… 125
5.1.2 利用预训练网络的模型压缩 …… 130
5.1.3 Mentee 网络与 FitNet … 134
5.1.4 使用预训练网络的应用：使用 CNN 的图像美学 … 136
5.2 生成网络 …… 138
5.2.1 自动编码器 …… 139
5.2.2 生成对抗网络 …… 142
本章小结 …… 147
参考文献 …… 148

附录 A Yann …… 152
后记 …… 164

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

第1章
Convolutional Neural Networks in Visual Computing

视觉计算简介

人们进行科学探索的目的是增强人类的能力。我们用火来烹饪食物，减轻了胃来加工食物的负担。这使得人类可消化的食物量增加，加速了文明的发展——这种文明水平是其他物种未达到的。轮子和交通工具的发明，可使我们旅行的速度大大加快。飞机的发明实现了人类遨游蓝天的梦想。人类发明和技术发展就是人类无止境地增强自己能力和探索未来的过程。

这些进步归功于人类的神经系统。人类错综复杂的神经系统具有思维、情感、推理、想象和哲学的能力。计算机视觉研究者偏向于人类视觉系统（human vision），但对我们来说，认知视觉系统（cognitive-visual system）是人类能力（包括智力）中最重要的部分。虽然人类视觉系统及其认知决策过程是很快速的，但并不是所有生物中最强大的，如敏锐性或夜视能力（Thorpe 等人，1996；Watamaniuk 和 Duchon，1992）。人类观察世界局限于一段很窄的电磁波谱，而许多其他生物的视觉感知范围比我们广得多。同时，人类也更容易产生许多视觉缺陷，如近视。基于上述考虑，我们很自然地想要改善我们的视觉能力，弥补缺陷。

多个世纪以来，我们发明了很多工具去超越人类肉眼的观察极限，如望远镜、双筒镜、显微镜和放大镜，去观察更远或更小的物体。无线电、

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

红外线和 X 射线设备可使我们观察到可见光谱以外的范围。最近，利用干涉仪发现了引力波，使得人类可用引力天文学来观察世界。虽然这些设备增强了人类的视觉能力，但学者和哲学家们早已意识到眼睛只是一种成像工具，真正“看到”的是大脑。

从柏拉图、亚里士多德、查拉卡、欧几里德到达·芬奇都研究了眼睛是如何观察世界的问题，但只有 Hermann von Helmholtz 在 1867 年所著的《生理光学》中首次提出：眼睛只捕捉图像，而大脑才真正“看到”和识别图像中的物体。在他的书中，他基于达·芬奇早期的工作，提出了关于深度和色彩感知、运动感知的新理论。尽管一直以来人们通过各种方式进行研究，Helmholtz 则首次提出了无意识推理（unconscious inference）的概念：不是所有的想法、思维和决定都是大脑有意识地完成的。Helmholtz 注意到人类很容易受到视觉错觉的影响，引用了太阳围绕地球旋转这个著名的误解案例，而事实上是地平线在移动。使用这种类比，Helmholtz 提出在人类无意识的情况下，是大脑在理解眼睛看到的图像，并推断和理解正在观察的物体。这可能是第一次深入洞察神经视觉的研究。一些近代早期的科学家（如 Blakemore 和 Campbell）开始争论一些现在已知的事实：大脑中有神经元负责估计物体的大小和方位感（Blakemore 和 Campbell，1969）。后来同时期的研究发现了人类视觉系统感知和识别颜色、形状、方向、深度，甚至物体的更复杂原理（Field 等人，1993；McCollough，1965；Campbell 和 Kulikowski，1966；Burton，1973）。

上述简短的历史叙述说明计算机视觉领域在丰富的人类技术发展中占有一席之地。本书重在简洁介绍现代计算机视觉技术，因为这项技术大多

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

来源于人工神经网络，所以称为“神经”计算机视觉（neural computer vision）。为了确保本书的完整性，我们在一些基础章节的开始处，通过定义基本概念、公式和方法，向读者介绍一般的视觉计算原理，之后的部分简要介绍图像表示。

1.1 图像表示基础

任何计算机视觉处理流程都始于成像系统，它从景物中捕获反射出来的光线，并将光信号转换成计算机可以读取和处理的图像格式。在计算成像技术发展的早期，图像通过把胶卷或印刷图像数字化后获得；而现在图像通常直接由数码相机获取，并存储成称为像素的有序数字。目前已有很多关于图像采集和照相机内部工作原理的教科书（例如光学、机械控制和彩色滤波等）（Jain，1989；Gonzalez 和 Woods，2002），因此对于这部分内容本书只做简单阐述。图 1-1 展示了光线进入相机形成图像的采样（即通过图像网格划分）和量化（即用有限的整数表示每个像素的颜色值）过程。

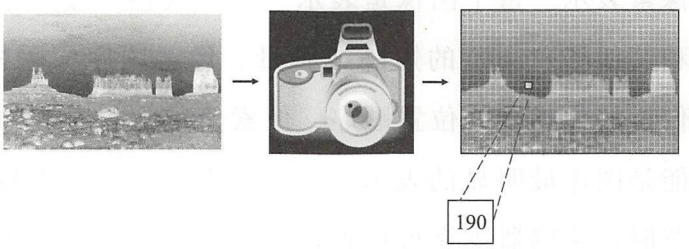


图 1-1 图像的采样和量化

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

所有图像可以看作一个矩阵（或三个矩阵，如果分别考虑色彩平面），该矩阵是景物的反射光在相机成像平面上的强度和色彩信息的量化数据。图 1-1 显示了景物在数码相机上成像的过程：相机中的传感器阵列决定了图像的大小和分辨率。假设传感器阵列有 $n \times m$ 个传感器，则产生 $n \times m$ 大小的图像。每个传感器捕获通过透镜入射到该传感器上的光线。对于一个 b 位图像，传感器为每个采样赋予 0 到（2^b-1）之间的值。假设一个 8 位图像，那么采样值就在 0 到 255 之间，如图 1-1 所示。上述过程称为采样和量化。所谓采样就是在连续视场中选择某些点；所谓量化就是把光强的取值限制在有限的离散数值内。在相机设计和相机模型中，采样、量化和图像形成涉及很多主题，感兴趣的读者可以查阅相关文献并进行更深入的研究。彩色图像在照相机中通常会分别对应生成红（R）、绿（G）和蓝（B）三个基色图像。如何产生这些 R、G 和 B 图像取决于相机，大多数消费级相机在传感器平面前设置滤色器来捕获所有三个颜色通道的马赛克图像，然后依靠“解马赛克”过程来创建全分辨率、分离的 R、G 和 B 图像。

有了这个装置，我们就能把图像表示为计算机中可存储的数字数据，称为图像的像素表示。每个图像是表示一个（灰色）或三个（彩色）或更多（深度和其他场）通道的矩阵或张量。像素顺序与采样顺序相同，也就是采集像素的传感器的位置顺序。像素值越大，对应的颜色强度就越大，这可能是图像最明显的表示方法。图像越大，像素越多。当要捕获景物的细节时，传感器需靠得更近，产生的图像分辨率也越高。如果用两个大小不同的图像来拍摄现实世界的相同区域和视场，那么较大图像比较小图像具有更高的分辨率，这是因为较大图像可以分辨更多的细

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

节。对于灰度图像，我们通常使用二维离散矩阵 $I(n_1, n_2)$ 来表示像素值矩阵，其中 n_1 和 n_2 分别索引矩阵的第 n_1 行和第 n_1 列的像素，$I(n_1, n_2)$ 的值对应像素强度。

虽然每个像素独立于其他像素进行采样，但相邻像素之间通常是相关的。因为典型场景不会在任何地方都有剧烈变化，除了两个不同实体间的边界像素外，相邻像素一般会很相似。因此，图像中像素值不连续（或变化较大）的“边界”能较好地识别出不同实体。通常情况下，除了实体边界的像素之外，自然景观的图像很多区域是平滑的（即没有变化或变化很小）。

如上所述，像素矩阵是图像的基本表示方式，通常称为空域表示。像素是对空间，更准确地说是对成像平面上的光强的测量值进行采样。还有使用所谓的频域方法来观察甚至获取图像，将图像分解成频率分量，类似于棱镜将太阳光分解成不同的色带。还有一些方法，如小波变换，使用时频变换来分析或分解图像，这里的时间在图像领域实际上是指空间（Meyer，1995）。所有这些表示方法都称为图像的变换域表示。通常，图像的变换域表示是可逆的，即可以从变换域表示返回到原始图像。实际上，对于特定的处理任务使用哪种表示确实是个问题。除了空间域和变换域中的表示之外，许多计算机视觉任务首先从图像中（原始图像或一些变换域表示）计算各种类型的特征，然后基于所计算的特征执行一些分析/推断任务。从某种意义上说，这种计算出的特征充当底层图像的新表示，故将此称为特征表示。下面，简要介绍几种常用的图像变换域表示和特征表示。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

1.1.1　变换域表示

也许被研究最多的图像（或任何序列数据）变换域表示是通过傅里叶分析（见 Stein 和 Shakarchi，2003）。所谓的傅里叶表示就是使用正弦函数的线性组合来表示信号。对于一个给定的图像 $I(n_1,n_2)$，可以用如下方式分解它（即逆傅里叶变换）：

$$I(n_1,n_2)=\frac{1}{mn}\sum_{u=0}^{n-1}\sum_{v=0}^{m-1}I_F(u,v)\mathrm{e}^{\mathrm{j}2\pi\left(\frac{un_1}{n}+\frac{vn_2}{m}\right)} \tag{1.1}$$

其中，$I_F(u,v)$ 是傅里叶系数，可以由如下方式（即傅里叶变换）得到：

$$I_F(u,v)=\sum_{n_1=0}^{n-1}\sum_{n_2=0}^{m-1}I(n_1,n_2)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{un_1}{n}+\frac{vn_2}{m}\right)} \tag{1.2}$$

在这种表示中，像素表示的图像 $I(n_1,n_2)$ 被分解成频率分量。每个频率分量的系数描述该频率分量存在的多少。频率分量在这里成为表示图像的基础。这种方法的常见应用是 JPEG（Joint Photographic Experts Group）图像压缩中用到的可变离散余弦变换（Discrete Cosine Transform，DCT）。JPEG 编解码器仅使用式（1.2）中正弦曲线的余弦分量，因此称为离散余弦基。DCT 基函数如图 1-2 所示。

从像素表示到变换域表示的任何变换核都可以写成 $b(n_1,n_2,u,v)$，逆向变换写成 $b'(n_1,n_2,u,v)$。对于许多变换，这些基在数学上通常是可逆的，相互之间可以转换。从图像空间到基空间的映射或变换可以公式化为：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

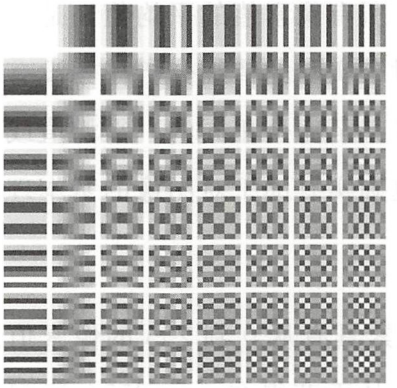


图 1-2 JPEG-DCT 基函数

$$I_T(u,v) = \sum_{n_1=0}^{n-1} \sum_{n_2=0}^{m-1} I(n_1,n_2) b(n_1,n_2,u,v) \tag{1.3}$$

它的逆变换为：

$$I(n_1,n_2) = \sum_{u=0}^{n-1} \sum_{v=0}^{m-1} I_T(u,v) b'(n_1,n_2,u,v) \tag{1.4}$$

式（1.3）是式（1.2）的一般化。许多图像表示可以通过这种形式在变换域建模，傅里叶变换只是其中的一种特殊情况。

1.1.2 图像的直方图

作为图像特征表示的第一个例子，我们讨论图像的直方图表示，它是给定图像的全局特征。首先考虑一个更简单的特征，由式（1.5）定义：

$$I_m = \frac{1}{nm} \sum_{n_1=0}^{n-1} \sum_{n_2=0}^{m-1} I(n_1,n_2) \tag{1.5}$$

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

该公式表示图像中所有像素值的平均值，是一个标量。这是对整个图像的一种通用表示，对于实际的精细分析任务没有多大用处，因为许多图像可能有非常相似的（甚至完全相同的）平均值或其他图像级特征。但这种简单的像素平均值也确实有助于对图像的基本理解，如从整体上判断图像是黑暗还是明亮的。

8 位图像的每个像素具有从 0 到 255 的像素值。分别对这 256 个值统计对应像素点的个数，从而获得在 [0，255] 区间上（仅考虑整数）像素密度的分布，以及与计数相对应的函数值（或像素的归一化计数），这就是直方图表示。对于彩色图像，可以很容易地扩展到三维直方图。图 1-3 展示了图像和对应的直方图。

一般来说，如果图像用 b 位表示，那么像素值具有 2^b 个量化级别，因此图像可由 0 至 2^b-1 范围内的值表示，那么现在图像的（归一化）直方图可表示为：

$$I_b(i)=\frac{1}{nm}\sum_{n_1=0}^{n-1}\sum_{n_2=0}^{m-1}\mathbb{I}(I(n_1,n_2)=i),i\in[0,1,\cdots,2^b-1] \quad (1.6)$$

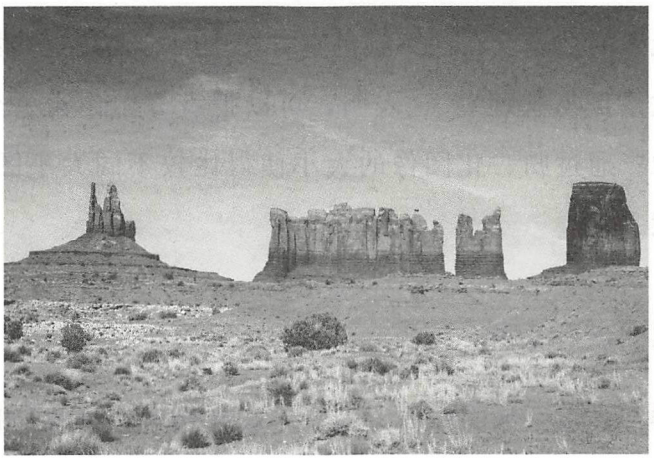
式（1.6）中，$\mathbb{I}$是一个指示函数，当其参数（视为逻辑表达式）为真时取值为 1，反之则为 0。公式用$\frac{1}{nm}$来标准化是一种消除特征与图像大小相关的简单方法。

虽然直方图比先前定义的平均值表示具有更多的信息，但仍然是非常粗糙的全局特征：视觉内容差别很大的图像可能会生成非常相似的直方图。例如，许多患者使用眼底成像仪进行视网膜扫描，尽管不同患者具有

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

明显不同的病理学图像，但它们可能生成非常相似的直方图。



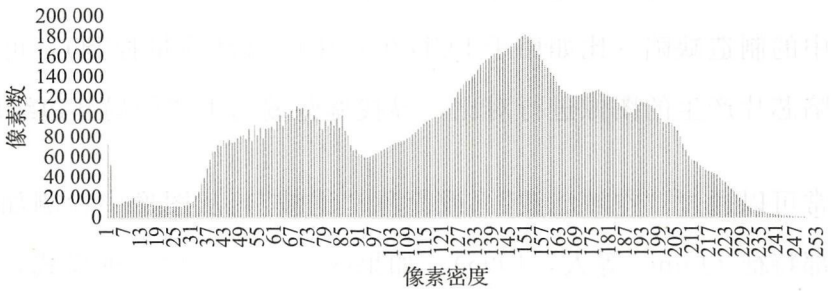


图 1-3 图像及其直方图

直方图也会丢失所有的空间信息。例如，给定一个直方图，代表绿色的条状提示有很多像素，但不能推断出这些绿色像素出现在图像中的哪个地方。但是训练有素的人（或机器）仍然能够从直方图推断出图像的一些全局属性。比如，如果我们观察到绿色条状上有许多像素值，就可能推断出是室外的图像，并且我们还希望能看到图像中有很多草。人们还可以从代表绿色的条状中推断出这幅图像可能是在斯堪的纳维亚农村拍摄的。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

直方图表示比原始图像要紧凑得多：对于一个 8 位图像，直方图实际上是 256 个元素的数组。可以通过进一步量化灰度级（如仅使用 32 个元素）来减少元素的数量（在计算颜色直方图中经常使用这一技巧，以使某些颜色条的数量更易管理）。这表明，可以把直方图作为一个良好的特征用于比较图像。通过归一化图像的大小使得比较不同大小图像成为可能，也变得更有意义。

1.1.3　图像梯度和边缘

当全局特征表示（如颜色直方图）不起作用时，许多应用需要局部特征。例如，如果我们试图通过计算机程序来识别液晶显示器（LCD）在显微图像中的制造缺陷（比如用于 LCD 生产中的自动质量控制），可能需要与无缺陷芯片产生的图像进行对比，寻找哪些偏离正常的局部边缘段。

通常可以通过“空域滤波”（或简称“滤波”）从图像中检测如边缘之类的局部特征（Jainet 等人，1995）。如果我们正在寻找一种模式，比如像素从较暗的区域从左到右快速变化到较亮的区域（存在垂直边缘），我们可以设计一种滤波器应用于图像，该滤波器将输出与输入大小相同的图像，对于存在急剧变化的像素赋予较高的值，而对于没有这种变化的像素赋予较低的值。这个响应值表示我们正在寻找模式的“强度”。滤波过程的实现可通过模板/掩膜（或简称滤波器）与底层图像 $I(n_1, n_2)$ 的“卷积”来实现。卷积的过程将在之后详细介绍。

上述寻找垂直边缘的例子中，可以将形式为 [−1，0，1] 的一维掩膜应用于图像的所有行。当与图像卷积时，该滤波器将产生另一个图像，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

其像素值（仅考虑幅度，因为滤波结果可能包含负数）表示给定位置周围的像素从左到右“上升”或“下降”的急剧变化程度。显然，使用上述滤波器的卷积相当于对 $I(n_1, n_2)$ 沿水平方向进行微分运算。因此，这个滤波器是一个梯度滤波器，它的输出是图像 $I(n_1, n_2)$ 的水平梯度。如果人们对二值边缘映射感兴趣，可以仅保留最强变化位置的像素，用滤波器对上述响应图像进行阈值处理就可以产生该结果。图 1-4 演示了将上述简单滤波器应用于 LCD 质量控制的图像处理技术。

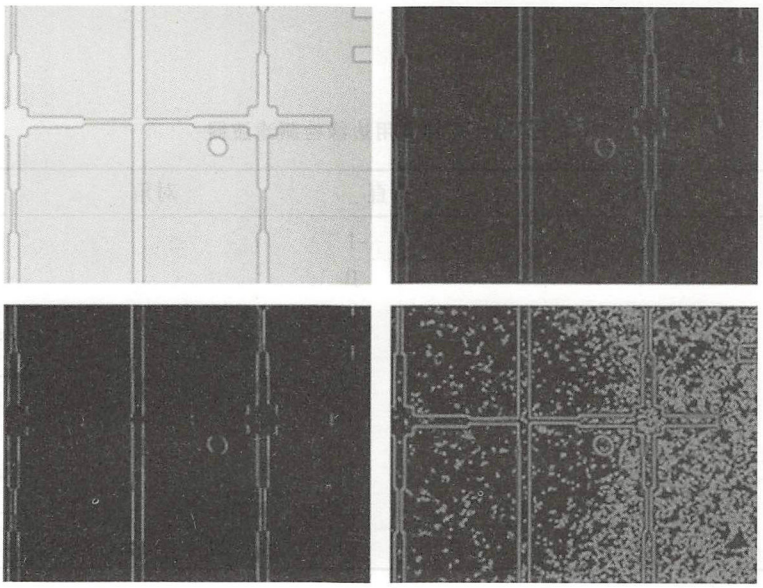


图 1-4 在第一行中，左边是显示 LCD 质量分析照片样本的细节图像；右边是同一图像与 [−1 0 1] 滤波器卷积后的输出图像。在下一行中，左边图像是来自先前卷积输出后再进行边缘阈值映射的输出图像，最后一个图像是 Canny 边缘检测器检测到伪边缘的输出图像。注意一维边缘检测器检测不到水平边缘，但 Canny 可以检测到这些边缘

上述简单滤波器可以转置以检测其他方向上的变化。此外，二维滤波器常用于图像，滤波器的大小也可以改变。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

表 1-1 总结了一些最早的边缘检测滤波器。虽然这些基本的滤波器可以产生图像梯度，这些梯度被阈值化后可以形成边缘图，但是这种简单方法得到的图像会有非常多的噪声。首先，确定一个好的阈值并不容易。而且来自自然图像的原始梯度值对于小邻域仍然可能保持大的强度值（而不是仅沿着细线变大），因此简单的阈值处理可能导致许多厚的边缘。因此，通常需要一些后续处理步骤来产生更局部化（薄）的边缘图，更好地与原始图像区域的感知边界保持一致。Canny 边缘检测器（Canny，1986）是一个非常好的常用后处理方法，其结果显示在图 1-4 中。

表 1-1　几种常用边缘检测滤波器

操作员	水平	垂直	对角	反对角
Roberts	$\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}$		
Prewitt	$\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 1\\ -1 & 0 & 1\\ -1 & -1 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}$
Sobel	$\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 2\\ -1 & 0 & 1\\ -2 & -1 & 0\end{bmatrix}$	$\begin{bmatrix}-2 & -1 & 1\\ -1 & 0 & 1\\ 0 & 1 & 2\end{bmatrix}$

现在将上述的滤波或卷积过程形式化。假设二维滤波器（或掩膜、模板、卷积核，在本文中可以互换）$F(l,w)$ 定义在 $[-a,a]\times[-b,b]$。为了方便，将过滤器的中心放在原点。滤波器与图像 $I(n_1,n_2)$ 卷积后产生输出图像 $Z(n_1,n_2)$：

$$Z(n_1,n_2)=\sum_{l=-a}^{a}\sum_{w=-b}^{b}F(l,w)I(n_1+l,n_2+w)\ \forall\, n_1,n_2 \tag{1.7}$$

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

注意，上述卷积应该产生与输入图像相同大小的输出图像。当处理图像边缘像素时需要注意一些问题，以便很好地支持卷积。为了确保卷积之后图像的大小保持不变，典型的方法是在卷积之前沿着边界填充必要的附加行/列。这种类型的卷积称为“same”卷积。如果不进行填充，则会得到一个“valid”卷积，其尺寸比原始图像小 $(a-1)\times(b-1)$ 个像素。二维滤波器 F 本身可以被认为是 $(2a+1)\times(2b+1)$ 的小图像。虽然方形滤波器是最常见的，但我们有意使用不同尺寸（a 和 b）的水平和垂直维度来说明使用非方形滤波器也是可以的。

目前讨论的梯度滤波器是基于图像 $I(n_1,n_2)$ 的一阶微分。拉普拉斯算子是二阶滤波器，它计算斜坡状边缘的二阶导数，结果表现为一对正负峰，从而将边缘定位为两个峰之间的过零点。数学上，二维拉普拉斯算子 $f(x,y)$ 可由下式定义：

$$f''_L=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2} \tag{1.8}$$

实现拉普拉斯算子的近似掩膜如表 1-2 所示。拉普拉斯算子是各向同性的，但由于它的二阶性质，易受噪声影响。因此，它常被应用于已通过一些降噪滤波平滑后的图像。

表 1-2 拉普拉斯算子

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

可利用小邻域内的像素平均来实现图像平滑。一个简单的均值滤波器

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

通常被称为盒状滤波器（box filter），其形式为：

$$B(u,v)=\frac{1}{U\times V}1\ \forall\, u\in[1,2,\cdots,U],v\in[1,2,\cdots,V] \tag{1.9}$$

其中，$U\times V$ 是滤波器的大小。例如，一个 3×3 的平均盒状滤波器的形式为 $\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$。该滤波器可以在 9 个像素的领域中平均像素。高斯平滑滤波器是利用高斯函数实现像素的加权平均：一个点离中心越远，则权重越低。高斯滤波器的形式如下：

$$G(u,v)=\frac{1}{2\pi\sigma^2}\mathrm{e}^{\left(-\frac{u^2+v^2}{\sigma^2}\right)} \tag{1.10}$$

加权平均除了在降噪中有用外，高斯平滑滤波器通常比盒状滤波器更受欢迎，因为高斯核通常用于建模离焦和成像平面上传感器单元的点扩散函数。高斯滤波器也可以被近似为 2D 卷积掩膜。典型的 3×3 高斯滤波器近似是 $\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$。高斯滤波器的优点是高斯函数的傅里叶表示是另一个高斯函数。由于卷积的效果很好，如果使用高斯函数就可以产生更快更高效的卷积运算。

高斯拉普拉斯算子（LoG）是应用于高斯滤波图像的拉普拉斯算子。高斯滤波器平滑图像，拉普拉斯算子检测平滑图像上的零交叉点。它检测到的点通常是隐藏着较好特征的关键点。由于卷积运算是关联的，所以可

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

以创建一个独立于图像的 LoG 滤波器。LoG 滤波器采用如下形式：

$$\mathrm{LoG}(u,v) = -\frac{1}{\pi\sigma^4}\left[1-\frac{u^2+v^2}{2\sigma^2}\right]e^{-\frac{u^2+v^2}{2\sigma^2}} \tag{1.11}$$

这实质上是高斯函数的二阶微分。LoG 也可以用卷积掩膜来近似，然后与图像进行卷积运算。图 1-5 是 LoG 掩膜的一个例子。

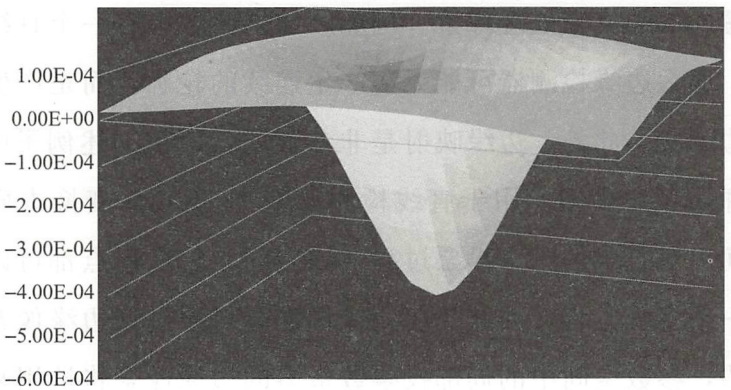


图 1-5 高斯拉普拉斯算子

1.1.4 超越图像梯度

图像梯度和边缘是最基本的图像特征表示，它们已经能支持一些推理任务。例如，可以将特定对象的模板边缘与给定图像的边缘进行比较，以检测和定位该对象是否出现在给定图像中（Li 等人，2001b）。然而，这些基本特征主要用于构建更复杂的特征，例如直线的 Hough 变换（Hough，1962）、Harris 角点（Harris 和 Stephens，1988），尺度不变的特征变换（Scale-Invariant Feature Transform，SIFT）（Lowe，1999）和面向梯度特征的直方图（HOG）（Dalal 和 Triggs，2005）。由于有大量关于此类特征

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

构建方法的参考文献，省略其内部工作的细节不会妨碍后期对神经网络计算机视觉的学习，所以我们仅在下面简要介绍。

1.2　基于 Hough 变换的直线检测

从自然图像中检测到的边缘常常是碎片化的，因为边缘检测的大多数方法只考虑局部邻域来处理像素。例如，如果图像中有一个直线形成的建筑物轮廓，那么边缘检测器可能无法产生连续的轮廓，而是产生轮廓上一些零碎的边缘段。此外，边缘映射是非参数的，即在上述例子中，没有包含任何轮廓的参数模型。用于直线检测的基本 Hough 变换本质上是在二维参数空间中对直线的一个投票过程：该空间中的任何点都可以用来定义图像中的一条直线方程，因此任何边缘像素会对通过该边缘像素的任何直线进行投票，参数空间中的局部投票数最多的边缘像素将被用来定义直线方程。虽然 Hough 变换的基本版本仅用于线形检测，但经过扩展后可检测其他形状（Ballard，1981）。

1.3　Harris 角点

Harris 角点，有时也需要“点特征”。例如，在估计两个图像之间的全局变换时，可能需要在给定的两个图像中找到一组匹配点。一个好的、有用的特征点应该具有唯一性。局部特征（如具有很大空间梯度的像素）可以通过前面讨论的梯度滤波器找到，但在实践中，许多这样的特征可能太相似以至于不能用于匹配任务。例如，沿着同一边缘段的像素可能

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

都具有相似的梯度，因此相互之间很难区分。从另一个角度看，若我们只通过一个小窗口检查直线/边缘，如果直线/边缘沿着它的轴移动，我们可能无法注意到运动（直到线的末端移动到可见窗口），这就是计算机视觉运动估计中的“孔径问题”。Harris 角点是基于上述观察得出的原理定义的：无论哪个方向，即使只通过一个小窗口观察，一个好的特征点应该是一个很明显的运动。视觉上，这种好的特征点通常对应于图像中的某种“角点”。在实现中，通过结构张量的特征值分析来检测角点，结构张量是由局部图像梯度导出的矩阵。

1.4 尺度不变的特征变换

为了使特征点在实践中有用，它应该对一些影响图像质量或外观的典型条件具有某种程度的不变性。例如，如果依靠某些特征点来匹配或跟踪不同照相机从不同视角获取的不同图像上的同一对象，我们要求这些特征对比例、旋转甚至外观的变化是不变的。

SIFT 是一种用于检测特征点（通常称为 SIFT 特征）的算法，这些特征点对尺度和旋转保持不变，或者仅对光照变化或局部几何失真会有轻微的外观变化。该算法首先创建原始图像的多尺度表示，然后找出高斯差分的极值，并将它们作为潜在特征点。有一些标准可用来删除被认为不好的潜在特征点（例如低对比度点）。最后，将为选出的特征点及其邻近区域计算描述符（descriptor），这个描述符（以及它在图像中的位置）基本上就是我们所说的 SIFT 特征。在 Lowe（1999）文献中描述了完整算法，还涵盖了基于 SIFT 特征的对象快速匹配相关的其他实现问题。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

1.5　方向梯度直方图

到目前为止，我们已经有了图像的局部和全局特征表示。局部直方图可以表示 $k \times k$ 块的直方图（$k < n$，m），描述在小区域块上的颜色、像素或边缘分布（Park 等人，2000）。Dalal 和 Triggs（2005）首次提出的 HOG 是一种流行的局部全局特征空间。给定一幅图像，我们很容易获得该图中所有像素的梯度。考虑将梯度方向量化为 j 个可能的区间。使用 $k \times k$ 的非重叠滑动窗口，我们可以遍历整个图像并创建每个小区域块的直方图，其中直方图的值是对应方向上梯度的大小。每个小区域块将有 j 个区间的直方图，每个区间（表示某个角度范围）表示该块中方向梯度的大小。HOG 特征是图像的形状表示，因为某些结构信息被保留在特征中。图 1-6 显示了这种表示的可视化。在每个块的这种特征表示中，箭头的长度描述了该箭头所指方向上的梯度大小。每个块具有 8 个区间的梯度直方图。可以看出，该表示显示边缘方向的方式优于普通边缘特征，不但显示了梯度的方向还显示了这些梯度的强度。

到目前为止，我们已经介绍了各种类型的滤波器，用于提取不同的特征。这些例子说明，滤波器可以被设计成任何可行的表示形式来有效地可视化图像。所谓特性就是我们想要测量数据的某个特定属性。我们需要做的是设计好的滤波器来获得好的特征表示。事实上，在卷积神经网络变得流行之前，计算机视觉的一个主要任务就是针对特定应用设计高效和易描述的特征表示。即使在今天，良好的特征表示也是将计算机视觉性能调高到令人难以置信的主要原因。现代神经计算机视觉依赖于

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

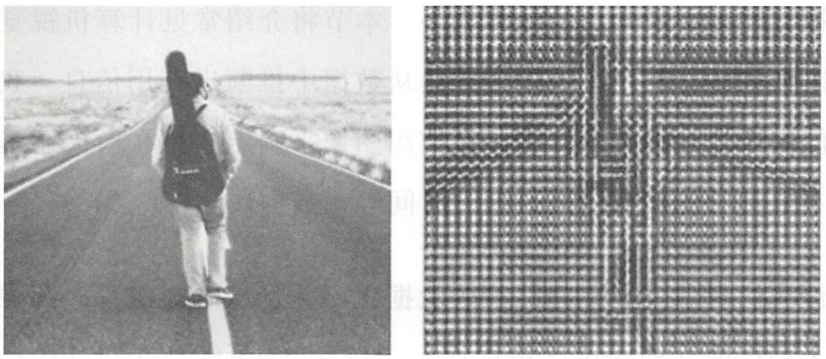


图 1-6 HOG 特征的可视化，使用 Hoggles 工具生成（来自 Vondrick 和 Carl 等人，“Hoggles：Visualizing object detection features”。*Proceedings of the IEEE International Conference on Computer Vision*，pp. 1-8，2013）

神经网络设计特定问题的特征。能自动设计图像特征的神经网络只是最近才流行，在进入自动学习特征的机器学习之前，研究和理解特征表示和机器学习的工作原理是有帮助的。

1.5.1 人工设计特征空间中的决策制定

我们已经回顾了在空域、变换域和特征域中表示图像的最常用方法。所有这些表示在各种计算机视觉应用中得到应用。尽管像素域中的基本矩阵表示也可以用于模板匹配、视频变化检测、区域分割等简单任务中，但更复杂的任务通常需要复杂的处理步骤，这些步骤通常需要在一些变换域或特征空间中操作。例如，考虑图像中的目标识别，通常需要先对图像进行一些特征表示（包括待识别对象的特征模型），然后采用一些决策过程来确定新图像是否应被归类到给定对象中（或不属于给定对象集）。决策过程通常基于机器学习方法，控制决策的参数通过数据样本的“训练集”（通常具有已知的标签信息）来确定。因此，机器学习方法已经成为许多

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

计算机视觉任务中最常用的技术之一。本节将介绍常见计算机视觉问题中的基本决策方法，利用统计学习方法从数据中推断出有用信息。我们从一个简单但有启发性的例子开始，只使用图像来确定它是在白天还是夜晚拍摄（即不依赖可能已存储的日期、时间等元数据）。

假设我们有白天和夜间图像的数据集，并且已经知道哪些图像是在夜间或白天拍摄的（即图像已被标记）。假设这些图像是在户外拍摄的（因此不存在人工光源控制图像亮度的情况），我们为每个图像提取一个非常简单的特征表示：所有像素的均值，如公式 1.5。一旦提取出所有图像的均值，就可以获得这些均值的直方图。又假设直方图可以用高斯概率分布建模，分别为白天图像和夜晚图像建模。图 1-7 展示了这种概率分布，横轴是从 0 到 255 的所有可能均值，纵轴表示具有特定均值的图像相对数量。虚线是夜间图像的直方图，实线是白天图像的直方图。从这个直方图中，我们可以直观地看到夜间图像通常比白天图像暗。此外，直方图有效地给出了图像取某一特定均值的概率估计。例如，夜间图像很可能具有 64 左右的图像均值（即虚线分布的峰值），而白天图像更可能具有 145 左右的均值。

假设现在有一个新的图像，它的拍摄时间未知，现在想估计一下。我们计算出这个图像的均值是 121（在图 1-7 中用星号表示）。使用从训练集中建立的分布，可以得出这样的结论：新图像更可能是白天拍摄的明亮图像。原因是均值为 121 的情况在训练数据集中白天图像数量是夜间图像数量的两倍多。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

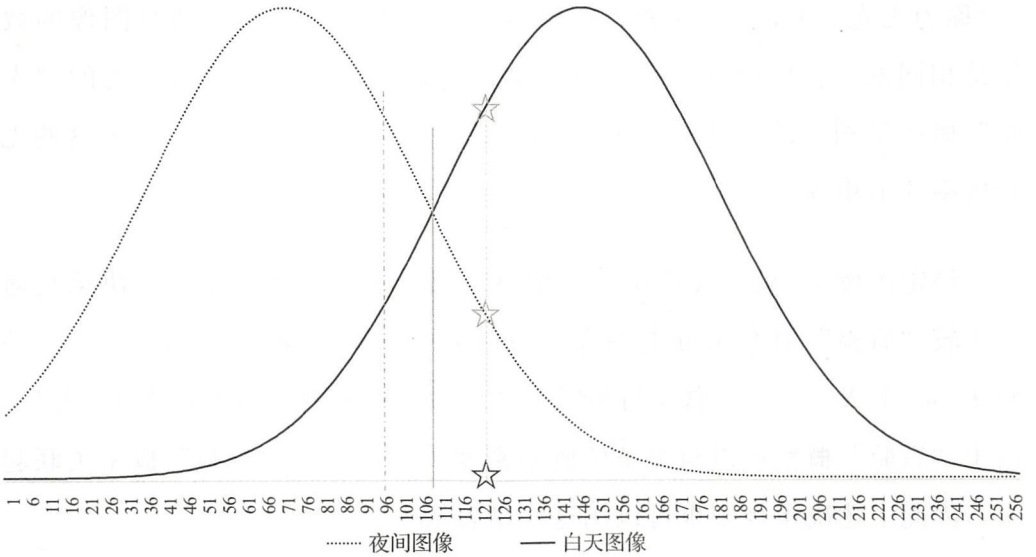


图 1-7 确定白天或夜间的图像。在星号上具有平均值的图像是白天图像而不是夜间图像的可能性更高。虚线是具有先验的决策边界，实线是没有先验的决策边界

1.5.2 贝叶斯决策

现在我们将上述直觉扩展为形式化的决策过程：贝叶斯分类。从概率上看，图 1-7 是两个高斯分布。我们用变量 i 来索引图像（均值表示），因此可以使用“图像 i”或简单的“i”来表示数据集中的第 i 个图像。我们用变量 ω 表示白天或夜间图像。如果 $\omega=0$，那么图像是夜间拍摄的；如果 $\omega=1$，那么图像是白天拍摄的。现在图 1.7 中的直方图可以解释为类条件概率密度函数（probability density functions，PDF）：虚线表示夜间图像的平均值 PDF，即 $p(x|\omega=0)$；同样地，实线表示 $p(x|\omega=1)$。我们称 $p(x|\omega=0)$ 和 $p(x|\omega=1)$ 为类条件 PDF（或简称为类条件）。我们进一步假设对于每幅图像而言，拍摄于白天或拍摄于夜间是等可能的

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

(也称为无先验假设)。或者，在给定的训练集中，如果这两组图像的数量是相同的，我们也可以得出这个结论。正规的表述是这两个类的“先验”概率是相等的：$P(\omega=1)=P(\omega=0)=0.5$。不过请注意，一般这些先验概率并不相等。

给定图像 j，就可计算出其均值 X。最佳（最小平均误差）决策是通过比较“后验”概率来分类图像（为白天或夜间图像）：$P(\omega=0|X=x)$ 和 $P(\omega=1|X=x)$。图像 j 将被赋予“后验”概率更大的类标签（0 或 1）。由于“后验”概率可以与类条件概率密度 $p(x|\omega)$ 和“先验”概率关联起来，因此上述决策规则也可以用这些量表示为：

$$\begin{aligned}&\omega=0,\ P(\omega=0)p(x\mid\omega=0)>P(\omega=1)p(x\mid\omega=1)\\&\omega=1,\ \text{其他}\end{aligned}\tag{1.12}$$

将“先验”纳入决策是贝叶斯方法的核心。在上面的例子中，因为人们更喜欢在白天拍摄图像，所以从数据集可发现白天拍摄的图像多于夜间，这个信息被纳入公式 1.12 的决策规则中。注意，当给出特征值 x 时，类条件概率 $p(x|\omega)$ 成为特定值 x 来自某个类的可能性度量，因此该类条件概率被视为（类的）“似然函数”（假设给定 x）。从这种观点看，先验是一种偏置（基于我们已有的先验知识），用来加权似然函数（基于我们从当前数据中可以学到的），以便在先验信息和当前信息之间取得平衡。

上述示例中，决策规则最终归结为寻找某个 x_d，它形成“决策边界”，最简单的一维特征空间中它仅是一个点。任何均值 $X<x_d$ 的图像将被归为夜间拍摄的图像，否则归为白天拍摄的图像。在图 1-7 中，当先验相等时，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$x_d=109$ 附近的决策边界将是最优解；而在 $P(\omega=1)>0.5$ 时，$x_d=95$ 是最优决策边界。

当我们能够在特征空间中对数据进行建模时，并且能够以某种方式获得类条件概率密度和先验概率分布时，贝叶斯决策过程是适用的。在这种情况下，可以得到如上所述的最优决策边界。在实践应用中，先验知识和类条件概率密度都需要从训练数据中估计。先验是标量，可以很容易从样本中每个类的相对频率估计出来。类条件概率密度需要用密度估计技术，一般有两种类型：参数估计和非参数估计。在前面的例子中，我们假设图像均值的概率密度函数服从高斯分布（即参数方法）。对密度估计感兴趣的读者可以参考 Duda（2012）等人的标准教科书。

1.5.3 线性决策边界

只要获得了所有必要的概率和密度，（在最小平均分类错误的意义上）贝叶斯决策就是完美的最优决策，但是对于许多实际应用，我们就不能可靠地估计出所需的这些值。为此，我们可以从简单的线性超平面入手，直接分离特征空间中的类。引入线性只是为了简化问题，理想/最佳的类边界一般不是线性的。在前面的例子中，问题就变成了寻找单点 x_d（无须依赖概率密度）的问题。如果有办法找出这个点，我们就可以做出决策。在线性方法中，该点通过特征的线性组合计算得出。例如，可使用下面的线性方法找到可能的 x_d：

$$x_d=\frac{1}{2}\left(\frac{1}{N_1}\sum_{i=1}^{N_1}x_i+\frac{1}{N_2}\sum_{i=1}^{N_2}x_i\right) \tag{1.13}$$

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

其中 N_1 和 N_2 分别是类 1 和类 2 的样本数。直觉上，对于决策阈值 x_d 的这种简单线性估计仅仅是两类图像均值的中间点。虽然没有理由相信这种解决方法是最优的（相对于贝叶斯决策方法的最优性保证），但我们可以获得线性假设所带来的简单性：只对样本进行了一些简单的加法（连同三个除法），获得的解与密度或概率无关。在实践中，我们不能保证上述解是最优的，因此定义解的更一般的线性形式：

$$x_d = \sum_{i=1}^{N_1+N_2} w_i x_i \tag{1.14}$$

然后尝试求出权重 w_i，以便得到更好的解。显然，式（1.13）是式（1.14）的特例。

上述基本思想可以推广到更高维的特征空间：对于两元分类，我们将一条直线作为二维空间的决策边界，三维空间就是一个平面，在更高维空间则为超平面。现在的任务是估计超平面的参数。有许多不同的技术可以实现，大多数技术都依赖于优化过程：首先定义目标函数（超平面参数的函数），其次寻求最优参数以最小化（或最大化，取决于公式）目标函数，通常有迭代方法（如采用梯度下降法）和一步法（如使用伪逆求解目标函数为最小均方误差的情形）这两种解法。

在所有线性方法中，最著名的可能是支持向量机（Support Vector Machine，SVM），其中所谓的“间隔”（margin）最大化，即决策超平面与最接近样本之间的距离最大化（Cortes 和 Vapnik，1995；Chapelle 等人，1999）。在基本的线性方法基础上可以往非线性方向扩展，其思想是将原始特征空间（非线性地）映射到新空间，然后在新空间中应用线性

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

方法，从而得到原始空间中的非线性决策边界。核 SVM 通过使用“核技巧”，不需要显式的特征映射就可以巧妙地实现这一目标（Bishop，2001）。

1.6 可变形零件模型的实例研究

到目前为止，所讨论的各种图像表示方法和决策技术似乎为我们提供了解决任何计算机视觉问题的必要手段。不幸的是，实际应用可能比我们目前所看到的例子复杂得多，例如识别非刚性物体。即使是前面介绍的那些高级特征，也可能无法充分描述不同结构和姿态的对象，更不用说支持部署 SVM 这样的决策算法了。我们现在把识别非刚性物体，例如识别图像中的人，说明如何在已介绍的基本表示/工具的基础上开发更多的方法。

考虑前面讨论的 HOG 特性。例如，如果想检测道路上的行人，我们可以创建一个图像的数据集，该数据集包含有用矩形边界框住行人的图像，也有行人不在矩形边界内的图像，或者框住的是非人类对象。我们希望从该数据集中学习到哪个是道路上的行人，哪个不是。我们提取所有边界框的 HOG 特征，如图 1-6 所示，然后利用这些特征来训练分类器。分类器可以是之前介绍的某类统计学习方法，或者是简单的模板匹配方案。

在这样做的时候，我们期望系统学会在特定姿势中检测具有特定结构的目标。这在某些情况下可能是有效的，例如直立人，但在某些情况下显然是不够的。可变形零件模型（DPM）在假设模型由子零件组成的情况下为对象创建模型。这些建议在过去曾多次提出，并以图结构（pictorial

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

structures）的名义进行修改完善（Fischler 和 Eschlager，1973；Felzenszwalb 和 Huttenlocher，2000）。子零件如前所述被检测为对象，主对象使用子零件构建。从数据集中我们只知道图像中用边界框表示的对象位置，但不知道它们的组成零件是什么。这些零件及其相对位置作为算法本身的一部分来学习。

DPM 使用两个组件工作：零件和弹簧。零件是每个零件本身的 HOG 模板，弹簧是零件相对于其他零件在特定位置的几何先验或成本。零件模型为零件本身创建匹配分数，弹簧确保整个对象可以用零件以有意义的方式表示。这在某种意义上与所谓的 Gestalt 原则是一致的，Gestalt 原则从各个部分来处理关于整体的推理（参见 Tuck，2010）。这使我们能够为许多不同的姿势和视角建模。

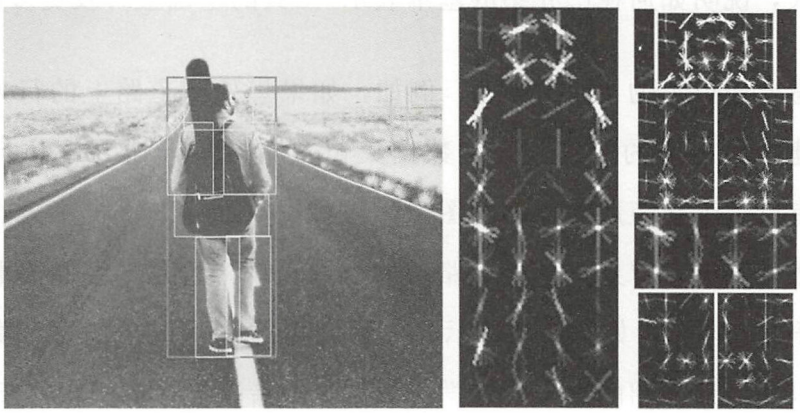


图 1-8　可变形零件模型。在左边，使用根滤波器和零件滤波器检测人，中间是人的根滤波器，右边是零件滤波器。这张图片是使用 DPM VOC 发行版 1 代码制作的。（来自 Felzenszwalb 和 Pedro F. 等人，级联对象检测与变形零件模型。计算机视觉与模式识别（CVPR），2010 IEEE 会议，IEEE，P2241～2248，2010 年）

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

DPM 已经有过几次升级更新。在深度 CNN 进入计算机视觉领域前，只使用建模而不是语法的 DPM 是最受欢迎的计算机视觉系统之一（Felzenszwalb 等人，2010）。DPM 为每个对象及其位置创建一个分数。这个分数是根分数的组合，它与 DalAl 和 TrimGS 提出的方法完全相同，即所有零件的匹配分数减去所有弹簧的分数（DalAl 和 TrimGs，2005）。可以用公式（1.15）描述：

$$s(p_1, p_2, \cdots, p_n) = \sum_{i=1}^{n} m_i(p_i) - \sum_{(i,j)\in E} d_{ij}(p_i, p_j) \tag{1.15}$$

第一项是所有零件的匹配分数，第二项是所有零件相对于彼此的成对弹簧分数之和。如果这个分数在所有零件组合（$p_1, p_2, \cdots, p_n$）中最大化，那么我们就命中匹配。该算法的实现使用动态规划来搜索所有这些零件，目前已经公开发布提供给公众使用。图 1-8 是使用这个公开算法包的例子。这样的对象建模比仅使用根模型更可靠，因此在许多最先进的对象检测基准数据集中获得了更高的性能。

1.7 计算机视觉向神经网络转变

到目前为止，我们已经研究了传统计算机视觉基本组成部分的一些基础知识。从前面的例子可以看出，一个传统计算机视觉处理流程包含定义一些特征表示（局部、全局或组合表示）和模式检测器/分类器（贝叶斯、支持向量机、基于模型的模式检测器，如可变形零件模型）。这种系统如图 1-9 所示。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

图 1-9　传统的计算机视觉系统

显然，在这样的处理流水线中，性能改进的创新机会在于寻找更好的特征表示，或者开发更好的分类算法，或者兼顾这两种，因为新的特征表示经常需要一些新的分类算法。对于给定的问题，依赖于显式的规则集来设计完美的特征和/或分类器是很有挑战性的，但另一方面，当有某种类型的训练数据可用时，许多计算机视觉研究者采用机器学习方法来开发新的方法。例如，在 Nagesh 和 Li（2009）提出了一种新的人脸识别表示方法，利用压缩感知理论（Candes 等人，2006）紧凑地表示同一主题的一组图像。Kulkarni 和 Li（2011）从大量图像块中学习到的仿射稀疏编码定义为图像特征，然后将这些特征用于图像分类。联合特征和分类学习的一个例子是 Zhang 和 Li（2010）提出的从标记过的人脸图像集中联合稀疏编码字典（即特征表示）和人脸图像稀疏表示的分类器进行学习。值得注意的是，许多流形学习技术在一些子空间中寻找更有效的数据表示，使分类特征可以做得更好，有助于上述处理流水线的特征提取阶段（Belkin 和 Niyogi，2001；Tenenbaum 等人，2000；Roweis 和 Saul，2000）。

尽管计算机视觉研究已有进步，但图 1-9 所示的流水线处理并非没有问题。一个突出的难题是，在大多数情况下，特征提取的显式设计依赖于应用和数据，因此设计者经常需要为新的应用人工设计新的特征，甚至需要为同一应用因新的数据类型重新设计特征。此外，人工设计特

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

征本身就是一个需要大量创造性、领域知识和经验的任务，因此不能指望任何人都能设计出接近最优解（如果存在的话）的方案（Venkatesan，2012）。

近年来出现了一类采用 CNN 来解决上述挑战的方法，大多和深度学习相关。人工深度神经网络是指一种学习方法，其中输入-输出映射的权重是根据网络构造的（LeCun，2015）。尽管神经网络通常学习的是分层结构和非线性映射，但学习权重实际上与我们先前介绍的线性分类器学习权重非常相似。由于 CNN 及其变体是本书的重点，我们推迟了对这类学习方法的讨论，将在后面的章节中进一步阐述。

本章前面，我们已经看到了用于梯度计算的卷积核或掩膜的例子。CNN 包含分类层和卷积层，其中卷积层采用的卷积核就是从数据中学习到的权重，从而在同一框架内有效地实现特征学习和分类器学习。很早之前就已发现多层 CNN 在解决具有挑战性的目标自动识别任务上具有很好的性能（Li 等人，1998，2001a），并且之前也尝试使用学习到的卷积核作为一般特征（Li，1999）。现在大量数据集的可用性以及计算能力的显著增加（与 15 到 20 年前相比）可为更具有挑战性的任务训练更深层次的网络。现在看来，神经计算机视觉已经进入到一个新时代，它采用数据驱动的方法不但可以学习分类器，而且还能学习特征表示，事实上，一些 CNN 模型已经被用作现成的特征检测器（Krizhevsky 等人，2012；Simonyan 和 Zisserman，2014；Szegedy 等人，2015；Soekhoe 等人，2016；LeCun，2015）。本书将通过深度神经网络带领读者探索现代神经计算机视觉方法。我们将在接下来的几章中介绍必要的基础知识。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

本章小结

人类的历史是以科技的发展不断提高人类的能力来标注的。在这种科技探索中，计算机视觉一直处于前沿，帮助我们在各种情况下更好地看到事物。尽管人工构造特征已经很流行，但最近基于深度神经网络的机器学习已经显示出很大的希望和潜力。本书将讨论利用 CNN 开发新的方法，它们极大地促进了计算机视觉的许多新突破。

参考文献

Ballard, Dana H. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* (Elsevier) 13: 111–122.

Belkin, Mikhail and Niyogi, Partha. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS Proceedings of the 14th International Conference on Neural Information Processing Systems*, 585–591, Vancouver, British Columbia, Canada.

Bishop, Christopher M. 2001. *Bishop Pattern Recognition and Machine Learning.* New York, NY: Springer.

Blakemore, Colin and Campbell, Frederick W. 1969. On the existence of neurones in the human visual system selectively sensitive to the orientation and size of retinal images. *Journal of Physiology* 203: 237.

Burton, Greg J. 1973. Evidence for non-linear response processes in the human visual system from measurements on the thresholds of spatial beat frequencies. *Vision Research* (Elsevier) 13: 1211–1225.

Campbell, Frederick W and Kulikowski, Janus J. 1966. Orientational selectivity of the human visual system. *Journal of Physiology* 187: 437.

Candes, Emmanuel J, Romberg, Justin K, and Tao, Terence. 2006. *Stable Signal Recovery from Incomplete and Inaccurate Measurements.* Hoboken, NJ: Wiley Online Library. 1207–1223.

Canny, John. 1986. A computational approach to edge detection. *Transactions on Pattern Analysis and Machine Intelligence* (IEEE) 8: 679–698.

Chandakkar, Parag S, Venkatesan, Ragav, and Li, Baoxin. 2013. Retrieving clin-

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

ically relevant diabetic retinopathy images using a multi-class multiple-instance framework. *SPIE Medical Imaging. International Society for Optics and Photonics*. 86700Q, Orlando, Florida.

Chapelle, Olivier, Haffner, Patrick, and Vapnik, Vladimir N. 1999. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks* (IEEE) 10: 1055–1064.

Cortes, Corinna and Vapnik, Vladimir. 1995. Support-vector networks. *Machine Learning* (Springer) 20: 273–297.

Dalal, Navneet and Triggs, Bill. 2005. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05). IEEE, pp. 886–893, San Diego, California.

Duda, Richard O, Hart, Peter E, and Stork, David G. 2012. *Pattern Classification*. Hoboken, NJ: John Wiley & Sons.

Felzenszwalb, Pedro F, Girshick, Ross B, and McAllester, David. 2010. Cascade object detection with deformable part models. *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2241–2248, San Francisco, California.

Felzenszwalb, Pedro F and Huttenlocher, Daniel P. 2000. Efficient matching of pictorial structures. *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings.* IEEE, pp. 66–73, Hilton Head, South Carolina.

Field, David J, Hayes, Anthony, and Hess, Robert F. 1993. Contour integration by the human visual system: Evidence for a local "association field". *Vision Research* (Elsevier) 33: 173–193.

Fischler, Martina A and Eschlager, Roberta A. 1973. The representation and matching of pictorial patterns. *IEEE Transactions on Computers* (IEEE) 22: 67–92.

Gonzalez, Rafael C and Woods, Richard E. 2002. *Digital Image Processing*. Upper Saddle River, NJ: Prentice Hall.

Harris, Chris and Stephens, Mike. 1988. A combined corner and edge detector. *Alvey Vision Conference* (Citeseer) 15: 50.

Hough, Paul VC. 1962. Method and means for recognizing complex patterns. US Patent 3,069,654.

Huo, Yuan-Kai, Wei, Gen, Zhang, Yu-Dong et al. 2010. An adaptive threshold for the Canny Operator of edge detection. *International Conference on Image Analysis and Signal Processing*. IEEE, pp. 371–374, Dallas, Texas.

Jain, Anil K. 1989. *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall.

Jain, Ramesh, Kasturi, Rangachar, and Schunck, Brian G. 1995. *Machine Vision* (Vol. 5). New York, NY: McGraw-Hill.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. Harrahs and Harveys, Lake Tahoe, NIPS, pp. 1097–1105.

Kulkarni, Naveen and Li, Baoxin. 2011. Discriminative affine sparse codes for image classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011*. IEEE, 1609–1616, Colorado Springs,

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Colorado..

LeCun, Yann. 2015. It's learning all the way down. *IEEE International Conference on Computer Vision.* Santiago.

Li, Baoxin, Chellappa, Rama, Zheng, Qinfen et al. 2001a. Empirical evaluation of FLIR-ATR algorithms–A comparative study. *Computer Vision and Image Understanding* (Elsevier) 84: 5–24.

Li, Baoxin, Chellappa, Rama, Zheng, Qinfen et al. 2001b. Model-based temporal object verification using video. *IEEE Transactions on Image Processing* (IEEE) 10: 897–908.

Li, Bao-Qing and Li, Baoxin. 1999. Building pattern classifiers using convolutional neural networks. *IEEE International Joint Conference on Neural Networks* 5: 3081–3085.

Li, Baoxin, Zheng, Qinfen, Der, Sandor Z et al. 1998. Experimental evaluation of neural, statistical, and model-based approaches to FLIR ATR. *International Society for Optics and Photonics Aerospace/Defense Sensing and Controls*. 388–397.

Lowe, David G. 1999. Object recognition from local scale-invariant features. *ICCV '99 Proceedings of the International Conference on Computer Vision.* IEEE, pp. 1150–1157, Fort Collins, Colorado.

McCollough, Celeste. 1965. Color adaptation of edge-detectors in the human visual system. *Science* (American Association for the Advancement of Science) 149: 1115–1116.

Meyer, Yves. 1995. *Wavelets and Operators.* Cambridge: Cambridge University Press.

Nagesh, Pradeep and Li, Baoxin. 2009. A compressive sensing approach for expression-invariant face recognition. *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.* IEEE, pp. 1518–1525, Miami, Florida.

Park, Dong Kwon, Jeon, Yoon Seok, and Won, Chee Sun. 2000. Efficient use of local edge histogram descriptor. *Proceedings of the 2000 ACM Workshops on Multimedia*. ACM, pp. 51–54, Los Angeles, California.

Roweis, Sam T and Saul, Lawrence K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* (American Association for the Advancement of Science) 290: 2323–2326.

Simonyan, Karen and Zisserman, Andrew. 2014. *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556.

Soekhoe, Deepak, van der Putten, Peter, and Plaat, Aske. 2016. *On the Impact of Data Set Size in Transfer Learning Using Deep Neural Networks*. Leiden, the Netherlands: Leiden University.

Stein, Elias and Shakarchi, Rami. 2003. *Fourier Analysis: An Introduction.* Princeton, NJ: Princeton University Press.

Szegedy, Christian, Liu, Wei, Jia, Yangqing et al. 2015. Going deeper with convolutions. *Proceedings of the IEEE, Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–9, Boston, Massachusetts.

Tenenbaum, Joshua B, De Silva, Vin, and Langford, John C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* (American Association for the Advancement of Science) 290: 2319–2323.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Thorpe, Simon, Fize, Denis, Marlot, Catherine et al. 1996. Speed of processing in the human visual system. *Nature* 381: 520–522.

Tuck, Michael. 2010. Gestalt Principles Applied in Design. Retrieved Nov. 30, 2015.

Venkatesan, Ragav, Parag, Chandakkar, Baoxin, Li, et al. 2012. Classification of diabetic retinopathy images using multi-class multiple-instance learning based on color correlogram features. *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, pp. 1462–1465, San Diego, California.

Von Helmholtz, Hermann. 1867. *Hanbuch Der Physiologischen Optik*. Scholar's Choice. Hamburg: Voss.

Vondrick, Carl, Khosla, Aditya, Malisiewicz, Tomasz et al. 2013. Hoggles: Visualizing object detection features. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, Sydney, Australia.

Watamaniuk, Scott NJ and Duchon, Andrew. 1992. The human visual system averages speed information. *Vision Research* (Elsevier) 32: 931–941.

Zhang, Qiang and Li, Baoxin. 2010. Discriminative K-SVD for dictionary learning in face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2691–2698, San Fransisco, California.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

第 2 章
Convolutional Neural Networks in Visual Computing

回归问题中的机器学习

第 1 章介绍了图像的基本表示并在此基础上进行推断（主要是分类）的技术。从前一章的讨论中可以看出，机器学习已经成为解决很多计算机视觉问题的关键技术。而计算机视觉问题中的许多机器学习任务最终可归结为一般的回归问题，其目标是找出从数据输入到输出的某些特定映射。在目标/图像分类中，我们可以有效地找到从特征向量映射到类标签的回归模型。回归模型也可以用来描述从基本的图像表示到其他抽象表示的映射，例如从原始像素值到实数特征向量的映射。第 1 章给出了一些具体映射例子，本章将用回归这个数学工具对这些映射进行完整的研究。

为了便于讨论一些关键思想，不被图像复杂的高维特征所困惑，我们创建一个简单原始的特征空间。我们首先讨论这个示例性特征空间的线性回归模型，并给出它的解析解，以及在基本模型中引入正则化的思想，然后讨论在基本线性模型上的扩展。由于存在解析解很难求出，甚至根本不可能求出的情况，本章的后半部分主要针对这种情况讨论迭代求解方法。

2.1 监督学习

第 1 章中研究的视觉特征虽然直观，但对于理解机器学习的一些基

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

本原理来说过于复杂。为此，我们创建一个更易理解的低维数据集来学习机器学习的一些基本原理。假设你是一家著名数据科学公司的人力资源经理，现在需要制定引进人才的策略。在面试候选人的时候，你想事先知道候选人的薪酬期望。在数据科学家需求旺盛的背景下，大多应聘者都有各自的薪酬预期。作为一名数据科学家，你可以建立一个模型预测应聘者的薪酬期望，并利用这些预测信息指导面试中的谈判。

一种直接方法是使用所有公司员工的薪酬来估计未来应聘对象的薪酬。假设你的公司有（$n+m$）名员工。如果获得当前员工的数据集，你可能会得出公式 2.1 的表示形式（仅考虑前 n 名员工作为训练数据）：

$$D=\begin{bmatrix} \boldsymbol{x}_1 & y_1 \\ \boldsymbol{x}_2 & y_2 \\ \vdots & \vdots \\ \boldsymbol{x}_n & y_n \end{bmatrix} \tag{2.1}$$

其中 $\boldsymbol{x}_i \in \mathbb{R}^d$ 是 D 维空间中的样本，每个维度对应于员工的某个属性的特定值，$y_i \in \mathbb{R}^1$ 表示员工的薪酬。员工的属性可以是雇员的工龄、职级或以前的薪酬等。剩下的 m 名员工的样本也具有类似的形式，在后期会使用到。$\boldsymbol{x}_i$ 通常也被称为特征、表示、样本、数据、预测变量、协变量、特性或属性，在本例中称为表示。表示空间或特征空间（本例情况下是 $\mathbb{R}^d$）包含所有可能的应聘对象的向量空间，每个应聘对象是该空间中的向量。每个样本都是该空间中的向量或点。类似于第 1 章中讨论的图像特征，这里描述样本的特征所使用的属性与我们的机器学习目的有关。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

当引用 y_i 时，通常称为变量、目标或标签，它们是我们要学习的目标。所谓学习就是要建立特征到标签的映射。为了建立员工薪酬的预测模型，先考虑特征空间是一维的情况 $x_i \in \mathbb{R}^1$，这个一维变量可能代表应聘者的工龄，如图 2-1 所示。

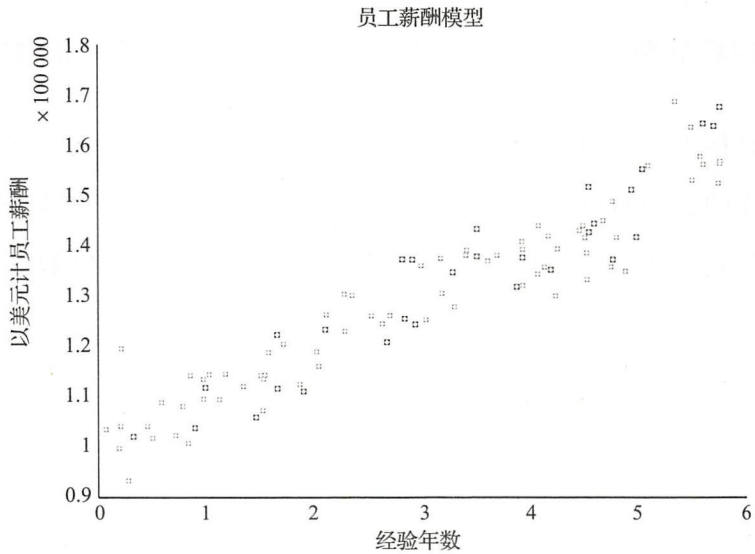


图 2-1　员工薪酬数据

监督学习就是利用训练数据 $(x_i, y_i) \in D(\forall i \in [1, n])$，用模型建立 x 与 y 之间的映射关系，当出现一个新样本 $x_{n+j} \notin D(j < m)$ 时，该模型可以给出它的估计 y_{n+j}。换句话说，我们用已获得的数据集的一部分来训练学习模型，余下的另一部分数据来测试模型。

我们把 D 称训练数据集，另一部分 $\{x_i\}(i \in (n, n+m))$ 称为泛化数据集。泛化数据集通常代表模型未知的真实世界。如果我们知道泛化数据集中的真实标签，那么这部分就可以用来做测试数据集，在部署到现场环

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

境前用它来评估模型的性能。模型其实是一种具有良好定义架构的函数变换，它将样本从特征空间映射到标签空间（label space）。

模型通常具有$\hat{y}=g(X,w)$的形式，其中w是将X转换为$\hat{y}$的参数或权重。从字面上看，X是行向量或列向量组成的矩阵，$\hat{y}$是预测列表，X中的每一行对应一个预测。我们使用这些符号来表示矩阵和向量。这里用$\hat{y}$而不是y是为了强调它只是模型的预测而不是真实的数据标签。简而言之，模型是预先定义的函数，它的参数需要从数据中学习得到，有时这种函数形式也被称为假设。

从数据集进行训练和测试的基本原理通常要求是不能从实际世界中动态地进行训练和测试。训练和测试数据集必须是稳定的，且尽可能与真实世界相似。这种相似性的正式定义就是概率分布。假设我们观察到的数据在真实世界服从某种概率分布。在本例中，应聘人员的薪酬要求和已雇员工的薪酬可能服从某些概率分布。我们假定已获取的数据的分布与真实世界即使不完全相同，也应该是相似的。这一假设在本例中是容易满足的，因为我们的数据集本身是从真实世界中抽样得到的。当数据集达到足够的抽样密度，就可以保证它有足够的准确率来模拟真实世界。如果数据集有无限个样本，该数据集就能代表真实世界。抽取的有监督训练样本越多，数据集就越可靠，从中学习到的模型也就越可靠（Button 等人，2013）。然而在机器学习中，我们只是试图找到数据集的某个拟合，希望该拟合是对真实世界的最佳拟合。在这种观点下，有监督机器学习就是通过抽样获得的训练数据集来对真实世界的概率分布进行最佳建模。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

2.2　线性模型

为了简化问题做出如下假设：假定应聘者的工作经验与薪酬是线性相关的，即应聘者的工作经验和薪酬之间的关系可以用直线表示。基于该假设，可将问题转化为线性回归问题。线性回归及其变体是数据分析中最常用的模型（Neter 等人，1996；Seber 和 Lee，2012；Galton，1894）。在数据和标签是线性相关的假设下，可建立一个线性模型 g：如果 $x \in \mathbb{R}^1$，那么

$$\hat{y} = g(x, \boldsymbol{w}) = w_1 x + b \tag{2.2}$$

如果 $x \in \mathbb{R}^d$，那么

$$\hat{y} = \sum_{i=1}^{d} w_i x^i + b \tag{2.3}$$

w_i 是线性模型 g 的参数，也表示拟合数据的超平面的法向量。习惯上，将 b 作为 w 的一部分，所以表示成 $b = w_0$ 且 $x^0 = 1$ 的形式。这样公式 2.3 就可用矩阵形式表示：

$$\hat{y} = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} \tag{2.4}$$

从图 2-1 中可以看出，没有工作经验的基本工资大约是 100 000 美元，工作经验每增加一年工资增加 10 000。有了这些信息，可以用 $w_1 = 10\ 000$ 和 $w_0 = 100\ 000$ 来进行建模。图 2-2 显示了用这些参数拟合数据的线性

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

模型。

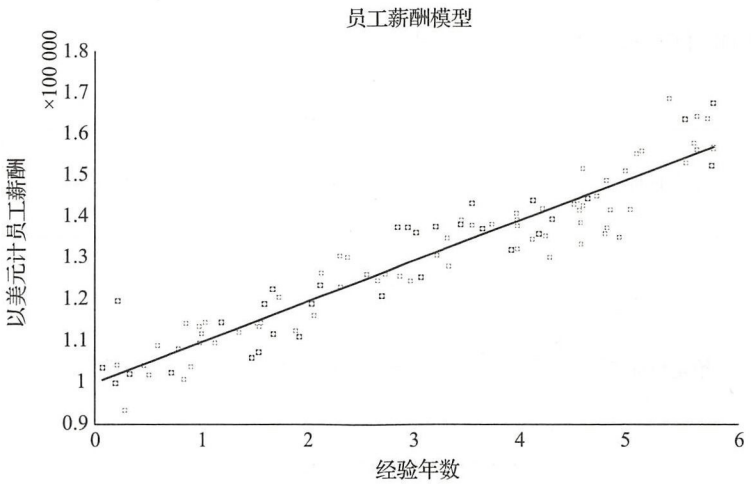


图 2-2 员工薪酬数据集的线性拟合

在二维数据集中观察直线（曲线）和估计模型参数（直线的斜率和截距）是一回事，更多的学习任务是在多维数据集中估计公式 2.4 的权重向量。为了能够将线性模型拟合到更大、更高维度的数据集，下面用更规范的方法来估计这些参数。估计这些参数以获得与数据集的良好拟合的过程称为训练。

给定一个数据集并假定用线性模型做拟合，但一开始并不知道该模型参数应该是多少。对于简单情况，可以利用线性方程组获得模型参数的唯一解。然而，对于大多数真实数据集，通常需要迭代方式求解：我们从解的一些初始猜测开始，然后迭代地改进，通常会用一些随机数来初始化模型参数。除非很凑巧，通常随机创建的模型肯定会产生错误的预测。在发现预测错误后，就可以基于这些错误预测来调整模型的参数，这样下一轮

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

迭代就会得到更好的预测。重复这个过程直到任何调整都不能改进对目标的预测。整个训练过程如图 2-3 所示。有了训练好的模型，我们就可以使用它对未知的标签进行预测（见图 2-4）。

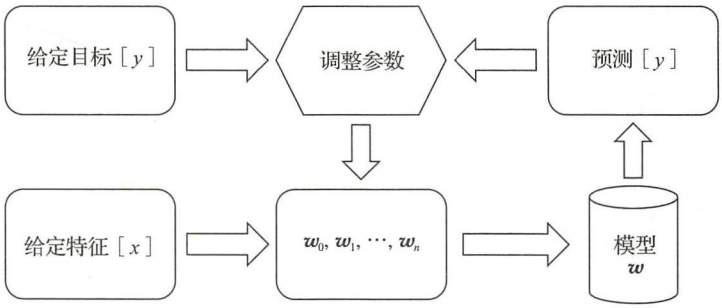


图 2-3　训练模型的示意图

图 2-4　利用模型预测标签的示意图

如果我们想通过参数的随机调整以得到更好的预测结果，这种思路显然是不够的。理想情况下，每次参数的调整都能改进前一次的预测。在本章的后面，我们将详细介绍这种参数调整方法，在下一节中我们将探讨一些估算模型参数的直接方法。

2.3　最小二乘法

从图 2-3 的训练步骤中可以发现：由于是从随机初始化权重 w 开始，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

所以预测标签$\hat{y}$也将是随机的。用一个误差函数e_i来衡量预测误差：

$$e_i(\boldsymbol{w}) = \| \hat{y}_i - y_i \|_2 \tag{2.5}$$

公式2.5表示在欧氏空间中，预测值$\hat{y}$，$\forall i \in [0,n]$与真实值y_i之间的误差。换句话说，给定一组模型参数$\boldsymbol{w}$，公式2.5定义的误差给出了模型预测的错误程度。上面的误差函数定义采用差值的平方，保证了误差是正值。

在图2-5中，实线是比任何虚线更好的模型。可以用以下方式来描述“更好”：将每个点的误差定义为点到直线的距离（公式2.6），这也是模型在预测该点时出现的误差，或者说是模型预测值与真实值的欧氏距离。因为实线具有比虚线更小的累积误差，所以实线是比虚线更好的模型。

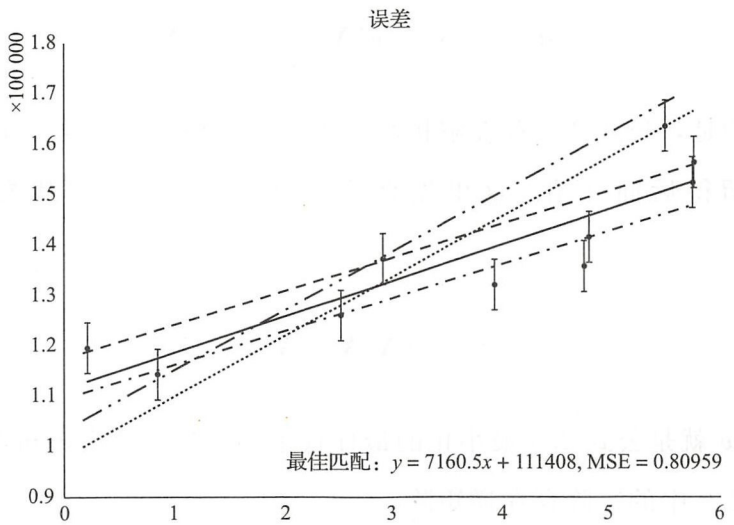


图2-5 可以看出，即使存在通过多个数据样本的其他行，实线也能产生与最小MSE的最佳匹配

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

即使实线没有直接通过任何样本，但是从累积误差看，实线比虚线具有更好的预测性能。这些虚线虽然至少通过了一个样本，但总体上它们具有更多的累积误差。整个训练数据集上的累积误差可以正式定义为：

$$e(\boldsymbol{w}) = \sum_{i=1}^{n} \| y_i - \hat{y}_i \|^2 \tag{2.6}$$

累积误差通常也可称为目标、误差、损失、代价或能量。我们现在的学习目标是找到误差 e 最小的模型，即最小二乘法的解（Lawson 和 Hanson，1995）。更正式地讲，希望参数 $\boldsymbol{w}=\ddot{\boldsymbol{w}}$ 满足

$$\ddot{\boldsymbol{w}} = \operatorname{argmin}_{w} e(\boldsymbol{w}) \tag{2.7}$$

按照我们之前的矩阵表示法，公式 2.7 可以写成：

$$e(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}) \tag{2.8}$$

幸运的是，公式 2.7 存在解析解。推导这个解析解需要一点线性代数的知识，留待后面介绍。这里先直接给出公式 2.7 的最优参数集的解析解：

$$\ddot{\boldsymbol{w}} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{2.9}$$

这里 $\ddot{\boldsymbol{w}}$ 就是公式 2.9 最小化的最佳权重参数集。这个解析解存在的前提是公式 2.9 中的矩阵存在逆矩阵。

尽管对于这类简单的线性模型确实存在解析解，但是对于更复杂的问题结构，必须依赖于在后面章节中介绍的优化过程。目前为止，我们仍然

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

采用这个解析解来研究线性回归问题。得到模型的参数解后，就可以对任何样本 $\boldsymbol{x}_{n+j}$ 用 $y_{n+j}=\hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_{n+j}$ 进行预测。这一预测过程称为测试，如图 2-4 所示。

2.4 极大似然估计的解释

让我们重新思考距离的概念。如果直线上的每个点都服从方差为 σ^2 的一维高斯分布，且模型训练充分（即能产生非常接近真实值的预测），则离直线一定距离处的高斯分布函数值描述了在该距离处找到点的概率。我们可以观察到，离这条线越远，找到样本的可能性就越小。高斯分布说明远离模型时找到实际样本的概率非常小，所以求解最小二乘法等价于给定训练样本的情况下，最大化模型的可能性。

图 2-6 描述了这种解释。我们可以看到用虚线连接的样本点到高斯分布函数的距离就是该样本点出现在该处的概率。注意，高斯分布的函数值都大于 0。样本离直线越远，其存在的概率就越小。基于这种解释，学习任务就是创建一条线段使得样本点在该直线上的高斯分布累积概率最大化。换句话说，也就是我们要选择使给定样本取得极大似然的模型。

进一步将上述观点公式化：

$$P(\boldsymbol{X},\boldsymbol{w})=\mathcal{N}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X},\sigma^2) \tag{2.10}$$

公式 2.10 是图 2-6 中线性回归的概率模型。高斯分布函数的均值就是我们的直线模型 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}$ 上的值。假设所有的样本是独立同分布的（i.i.d），

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

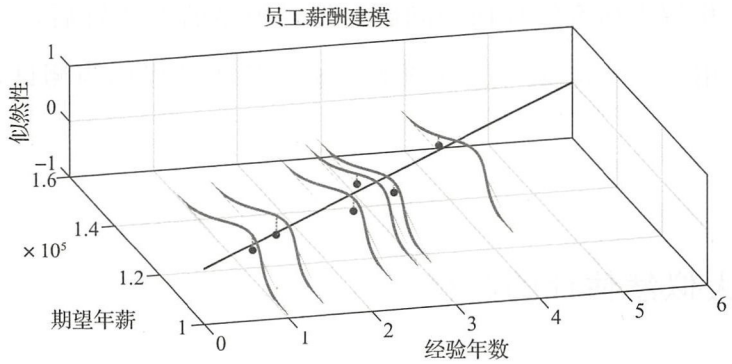


图 2-6　观察样本的概率用到真实模型线上的高斯分布函数的距离给出

我们可以将公式 2.10 重写为

$$P(\boldsymbol{X},\boldsymbol{w}) = \prod_{i=1}^{n} \mathcal{N}(\boldsymbol{w}^{\mathrm{T}} x_i, \sigma^2) \tag{2.11}$$

我们可以进一步把公式 2.11 改写成对数似然：在给定一条直线的情况下（这条直线决定了高斯分布的均值），这些数据点存在的似然对数为：

$$\log P(\boldsymbol{X}|\boldsymbol{w}) = \sum_{i=1}^{n} \log P(y_i|\boldsymbol{x}_i,\boldsymbol{w}) \tag{2.12}$$

最大化对数似然是机器学习中的常用手段（Nadaraya，1964）。似然在我们的例子中就是指找到离模型更近的那些样本的累积概率。由于对数是单调函数，因此最大化似然与最大化对数似然求得的最优解是相同的。与最佳拟合模型不同的是，极大似然估计（MLE）是试图找到最大化数据集存在可能性的模型。

由于我们的模型假设服从高斯分布，因此极大对数似然函数可进一步表示为：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$$l(\boldsymbol{w}) = \sum_{i=1}^{n} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^2 \exp\left(-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)\right)\right] \tag{2.13}$$

一个有意思的发现是，可以证明最大化公式 2.13 等价于最小化公式 2.14，且与公式 2.7 完全等价：

$$e = \sum_{i=1}^{n} (y_i - \boldsymbol{w}^{\mathrm{T}}x_i)^2 \tag{2.14}$$

使用线性回归模型进行预测其实是对该点的变量间是否服从线性关系进行评估，而使用 MLE 进行预测则是对该点是否服从高斯分布进行评估。使用 MLE 的优点是除了具备预测能力之外，还可以获得该预测的置信度（高斯分布函数值）。

2.5 扩展到非线性模型

到目前为止，我们看到的模型只是一条简单的直线。在本节中，将讨论线性假设可能不成立的情况。由于我们对线性模型很满意，所以当数据集无法用线性模型很好地建模时，就设法将数据集转换成线性模型可以工作的样子。这种方法包括其他的核方法是计算机视觉领域中除神经网络以外最流行的方法之一（Shawe-Taylor 和 Cristianini，2004；Ramsay 和 Silverman，2006；Vapnik 等人，1997；Gunn，1998）。

将线性模型扩展到非线性模型可以简单地通过用 $\boldsymbol{w}^{\mathrm{T}}\Phi(\boldsymbol{X})$ 替换 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}$ 来完成，其中 $\Phi(\boldsymbol{X})$ 是 $\boldsymbol{X}$ 的非线性映射。例如，执行如下回归：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$$\hat{y} = w_1 \sqrt[2]{x} + w_2 x + b \tag{2.15}$$

其实是在执行 $\Phi(x)=[\sqrt[2]{x}, x]$ 变换后的线性回归。尽管通过引入 $\Phi(x)$ 可实现线性回归，但 x 和 y 之间不再是线性关系。这种非线性变换称为基函数展开。

利用基函数，我们将数据映射到一个新的空间，在该空间上我们有望获得预测性能较好的线性模型。在后面的章节中，我们转向研究神经网络时，将会自动学习得到变换函数 Φ。但在现阶段，我们使用显式高阶模型进行预测，但在任意模型复杂度上执行回归会出现很多问题。

第一个问题是在图 2-7 中出现的现象。模型的阶数越高，拟合曲线就越复杂，能拟合到的样本点也就越多。拟合更多的样本点听起来是个不错的方案，但通常不能使累积误差达到最小值，这在图 2-5 中已经证实了。

所有这些扭曲和转折的曲线意味着对于一些测试数据，可以预测很多不同的值。例如，对于图 2-7 中标记为★的这个数据点，如果我们使用线性模型而不是六阶多项式模型，那么员工的薪酬就可以用金钱来正确预测。如果用六阶多项式拟合，预测值就会低于应聘者的薪酬期望，导致他们拒绝接受公司的录用。这个问题被称为过拟合（overfitting）。过拟合是许多学习系统部署失败的最常见原因之一，表现为在训练数据集上产生非常好的预测效果，但在测试数据集中却表现出很差的预测性能。过拟合通常很容易被发现。如果模型对一些样本的预测极其准确，而对其他样本的预测非常糟糕，就极有可能出现了过拟合问题。如果在训练集上有非凡的表现，在测试集上性能很差，那也很可能出现了过拟合。避免过拟合的一

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

个简单方法就是正则化。

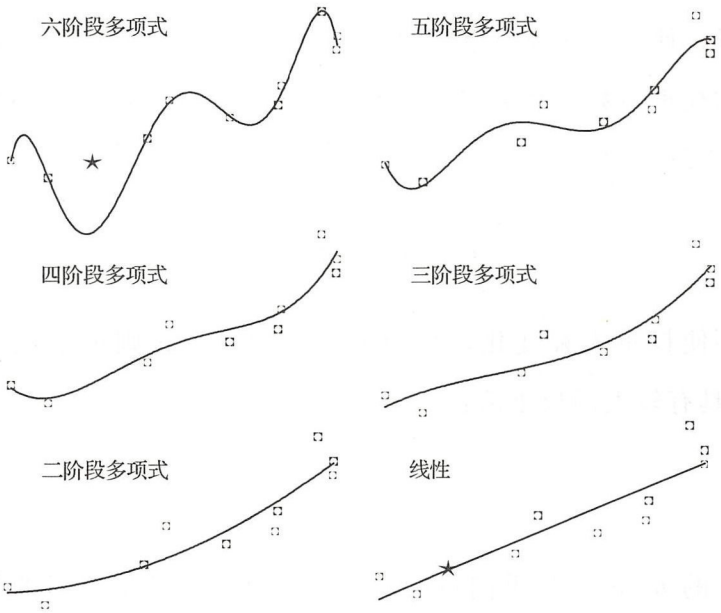


图 2-7 使用不同阶的多项式基函数实现线性回归。注意，这里所示的模型是建立在原始向量空间上的，因此是非线性的。事实上它们是在变换的特征空间上学到的线性模型，其中特征因不同的顺序会有不同的位置，所以在变换空间中无法进行有意义的比较。因此，我们把变换空间上的线性模型重新逆映射到原始空间上，从而更容易比较这些非线性模型

2.6 正则化

如果曲线扭转变化得非常复杂，则该曲线的系数（权重）也必然在很大的正负值区间内大幅变化。为了避免曲线的这种复杂扭曲，从而防止过拟合，我们可以对损失函数增加惩罚项。在正则化学习中，有很多不同的

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

方法来增加惩罚项到误差函数。首先阐述下增加惩罚的理念。当我们试图最小化（或最大化）某个函数时，可以在损失函数上添加一个想要限制它的加权度量，使它服从我们想要的某些约束和特性。现在使用这个直觉来约束模型的权重系数，避免它们取太大的值。将我们试图最小化的误差函数公式 2.14 改写为

$$e = \| y - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \| \tag{2.16}$$

为了不使权重大幅变化，我们可以添加一个正则项来惩罚，保证公式 2.17 不具有较大的权重值：

$$e = \| y - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \| + \alpha_2 \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} \tag{2.17}$$

上式中的 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ 是权重向量的长度。机器学习的最终目的是减少误差 e，我们希望最小化添加到误差 e 上的权重向量长度 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$。如果我们添加的是常量或者是与数据集的标签预测无关的内容，则不会影响学习过程。在公式 2.17 中，我们添加了一个关于权重的函数到误差函数中，这就会迫使在学习过程中检查权重值是否过大。

在这种考虑下，正则项就不应该包含 w_0，因为函数的截距不影响曲线的复杂性。α_2 是平衡损失函数值和惩罚项的因子，这样就使得一个损失不会被另一个损失的幅度所抵消。α_2 取较大值时，就会迫使权重取较小值，但对误差影响不大；而 α_2 取较小的值时则起相反的作用。α_2 本质上是在学习器在乎的那些方面做出折中。这种惩罚策略可确保较小的权重，同时误差也被最小化。因此最小化公式 2.17 中的函数是在取得更好的拟合（第一项）和更小的权重（第二项）之间做出折中。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

可以证明，我们在线性回归的损失函数中加入惩罚项后，仍然可以获得解析解：

$$\boldsymbol{w} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \alpha_2 \boldsymbol{I}_d)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{2.18}$$

其中 $\boldsymbol{I}_d$ 是 d 维单位矩阵。这个解析解有一个很好的性质，即它的逆矩阵比公式 2.9 中的逆矩阵具有更好的数值稳定性，但通常得不到比简单线性回归更好的解。

这个过程被称为正则化，因为它限制住了函数中的权重向量的扩张。这种特殊的正则化被称为 L_2 正则化，因为我们用权重向量的 L_2 范数来惩罚损失函数。而这种带 L_2 正则化的回归也常被称为岭回归（Hoerl 和 Kennard，1970；Le Cessie 和 Van Houwelingen，1992；Marquardtand Snee，1975）。

此外，我们也可以把降低模型本身的复杂性看作是另一种正则化。如前面的例子那样，如果试图用六阶多项式做拟合，在保持性能的前提下，我们倾向选择一个更小的模型。可以通过惩罚非零权重的数量来执行这一点，而不是像L_2 正则化的情况那样惩罚权重的变化幅度。非零系数越多，模型的阶数就越高，因此可用L_1 范数代替L_2 范数来实现这一点：

$$e = \| y - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \| + \alpha_1 |\boldsymbol{w}|_1 \tag{2.19}$$

这种正则化通常被称为L_1 正则化，它可以保证我们得到一组权重的稀疏解。为了使公式 2.19 最小化，我们需要使第二项尽可能的小。降低第二项的唯一方法是使某些权重系数降到零。通过强制权重系数变为零，就可实现它们之间的稀疏性。得到一组稀疏的权重系数意味着，所有的权重

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

系数组合成的向量，其中大多数分量将是绝对零值，这是一种非常激进的正则化。

一旦我们拥有稀疏的权重，相当于选择哪些属性参与曲线拟合，哪些不需要。因此，通过权重稀疏化，我们也实现了特征选择。L_1 正则化可以帮助我们从极其复杂的模型开始，然后进行简单的正则化，在保持良好的泛化性能的前提下去掉尽可能多的特征。

正则化为什么能起作用？对于一个问题，用更复杂的解决方案一定有用吗？奥卡姆剃刀的简约主义哲学简洁地回答了这个问题（Jefferys 和 Burger，1992）。它声称最简单的解决方案往往是最准确的。更确切地说，奥卡姆剃刀定律就是在相互竞争的推断中，应该选择假设最少的推断。正则化是从外部阻止非常复杂的模型，并消除尽可能多的变量参与到预测中。对世界的极端解释，即使符合我们观察到的数据，也不总是正确的解释。极端解释和复杂模型通常不能很好地推广到未来的测试数据。他们也许能很好地处理几个样本，但这可能只是因为特定训练样本的特异性质决定的。我们宁可选择略低于整个数据集的拟合度的模型，也不选择完美拟合少数样本特性的模型。

2.7 交叉验证

我们已经远远超出了最初仅仅调整线性模型系数的层面，而是变成调整整个系统的工作。在这个系统中，我们需要选择许多超参数（hyper-parameters），包括模型大小、正则项的权重和类型。尽管权重系数可以从

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

数据中学到，但是与预测不直接相关的其他模型参数必须由人工设定，如正则项的权重。为了准确地选择这些超参数，我们可以巧妙地利用原始训练数据本身。

我们将原始的训练数据分成两部分：验证集和训练集。假设我们选择一个超参数（比如 α_1），我们可以先固定 α_1 的某个值，然后在训练集上进行训练，最后在验证集上评估模型的性能。在验证集上取得最佳性能的设置就是最好的参数设置。在这种情况下，验证集与训练集应该不相交，且保持一致性。验证集通常小于训练集且与测试集具有相同的概率分布。

我们可以用原始训练数据的不同部分作为训练集来创建不同的模型，同时它们又可作为其他模型的验证集，最终的预测是所有这些模型预测的平均。这种交叉验证的方式被称为 k-折交叉验证（Kohavi，1995；Efron 和 Gong，1983）。交叉验证技术对岭回归中确定超参数方面的发展具有特别重要的影响，后来该技术被应用到其他线性模型，并被进一步推广到一般的模型选择问题上（Golub 等人，1979；Shao，1993；Kohavi，1995）。

图 2-8 显示了 3-折交叉验证的设置过程。将原始训练数据划分为三个不相交的分组，接着将第一组作为验证集，其余组作为训练集；然后循环使用此策略，直到三个组都作为验证集被使用过为止。最终得到 $k=3$ 个模型，我们的最终预测是这三个模型的平均。交叉验证的极端情况是留一（leave-one-out）交叉验证（Zhang，1993）。如果原始训练数据有 n 个样本，我们将 $n-1$ 个样本作为训练集，余下的 1 个样本为验证集。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

图 2-8　交叉验证

2.8　梯度下降

前一节研究了线性回归模型的建立，主要关注解析解，并意留了未讨论的另一种解法：如何迭代调整权重以获得最小的误差。这种训练过程是一种通过正确的方式旋转权重向量，以实现误差的最小化的数学过程，通常是基于梯度的优化技术。

之前讨论了线性回归问题的解析解，但是在我们写出解析解之前，可以反复尝试随机旋转权重向量，直到在测试集中取得满意的性能（Boyd 和 Vandenberghe，2004）。事实上，获得解析解对于很多实际的线性回归问题是不可行的。为了计算解析解中的矩阵逆或伪逆需要大量的内存和运算时间，同时还需要保证逆矩阵是稳定的和非奇异的。在后面的章节中会看到，解析仅适用于线性回归模型。如果面对更复杂的模型，如增加几层非线性映射，得到的解析只能凭运气了。

第 1 章中描述的数据是静止不变的，这对于公司员工的薪酬是可能的，但在现实中的其他地方却很少成立。真实世界的数据通常是动态变化

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

的且是以数据流形式产生的。即使数据不是自然产生的数据流，也会因为计算机内存的不足而只能分批处理。解析解需要一次读入所有数据完成计算处理，不能采用数据流或批处理形式。此外，对于前述的解析解，如果数据发生变化，则需要重新求解权重，因为旧的解析解不再是最优解了。

基于上述讨论，本节中我们将寻找求解 $\dot{w}$ 的实用方法。在求解过程中，我们还将发现所得参数的独特性与偶然性。回顾前一节中创建的误差函数：

$$e(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}) \tag{2.20}$$

这个误差函数在参数空间中呈现碗的形状。图 2-9 显示了误差函数在二维参数空间的可视化效果，其中 w_0 和 w_1 是模型的参数。此平面上任何一点都代表了一个特定模型的参数组合。这些特定的参数组合对应一个已被一组具体的权重值确定的回归模型。

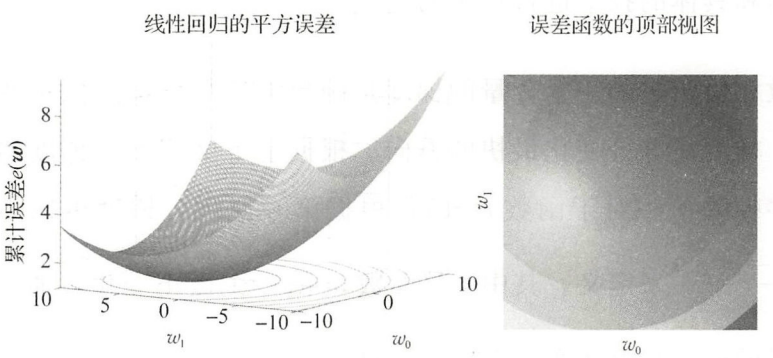


图 2-9 碗形的误差函数。左边是一个三维可视化，它的等值线投影在参数平面上。右边是同一个函数的俯视图。在这两个图中，等值线的圆周是具有相同损失值的点的轨迹。沿圆周运动不影响损失；沿垂直于切线方向向内移动是函数值下降最快的方向

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

在 w_0 和 w_1 的空间中通常会存在使误差最小的参数组合，本例中是 $\ddot{w}=[\ddot{w}_0,\ddot{w}_1]=[3,3]$，它是一个双参数的线性回归模型。这里需注意的是，我们假设特性和要学习的输出是线性相关的，只有这个线性假设成立，回归模型的解才会出现在这个参数空间中。找到的解也许在训练数据或测试数据上表现得不是很好，但至少存在一个最优解。在机器学习和计算机视觉文献中经常使用术语“假设”，指的是模型的类型或函数形式。给定不同的假设，例如两次基函数展开，可能比线性假设好，也可能不好。

假设我们建立的是参数为 w_0 和 w_1 的双参数线性模型，那么最优解是 $[\ddot{w}_0,\ddot{w}_1]=[3,3]$，如图 2-9 所示。也就是说，给定假设，那么最优解就是所有可能的参数组合中产生最小误差的参数组合。

在前几节讨论的机器学习训练过程中，我们从一个随机参数集开始旋转权重向量，直到得到最优解。在接下来的章节中，我们将会建立一个正式的过程和具体的技术进行权重的迭代搜索。

函数相对于它的一个分量的梯度是函数随这个分量的微小变化而变化的速率。梯度是该处变化最快的方向。根据上下文假设，梯度的量化形式可采用牛顿差值。对于函数 $e(w)$，根据第一原理，梯度可以由 $e'(\boldsymbol{w})=\dfrac{e(\boldsymbol{w}+\varepsilon)-e(\boldsymbol{w})}{\varepsilon}$ 来定义，其中 ε 是参数 $\boldsymbol{w}$ 的一个微小变化。通过用 ε 扰动某个位置的 $\boldsymbol{w}$，观察其变化来估计该处的梯度。

通常，我们没必要依赖第一原理的扰动行为来获得梯度。大多数回归和神经网络都有它们封闭形式的梯度。例如，公式 2.8 的梯度是：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$$e'(\boldsymbol{w}) = -2\boldsymbol{X}^{\mathrm{T}}Y + 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w} \tag{2.21}$$

注意，上面的梯度是由逐个分量组成的向量。该梯度向量的分量就是权重分量变化对误差影响的程度。

假设图 2-9 所示的碗状函数是我们的一个错误函数。除了碗底的点外，任何曲面上点都会有方向指向外面并远离碗底的梯度。梯度方向就是函数上升最快的方向。因为我们的目标是最小化误差，所以不能沿着梯度方向去改变权重，而应该从相反的方向上去改变权重。我们要找的最优权重系数就是误差函数的最低点所在的位置，该处梯度始终为零。如果我们发现某个权重系数组合中，不管朝哪个方向看，梯度方向都指向外，大小呈上升趋势，这说明已达到了误差最小位置。从另一种角度看，函数某点上的梯度是该点的曲率或斜率的度量，其最小值的点就是斜率为零的点。

现在考虑如下过程：我们从随机初始化模型参数开始，这些随机参数对应于 w 空间中的某一点。除非我们非常幸运，随机参数刚好在碗的中心位置，否则总有降低误差函数的空间。当参数位于 w 空间中的某个位置时，从该点周围寻找最陡下降方向，就是该处梯度方向的相反方向。

我们朝这个方向前进一小步到达一个新的位置，计算上就是从权重分量中减去这一小步的对应分量。我们重新计算这个新位置的梯度，决定下一步的前进方向。除非该处的曲率不再减小，否则将继续重复这一过程。如果我们发现处于一个曲率不再向任何方向减小的点，说明已经到达了谷底。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

上述思想发展出了一种迭代学习算法：梯度下降。更正式地表达为，假设第 τ 次迭代到某个位置 w^{τ}，我们以如下方式更新权重：

$$w^{\tau+1} = w^{\tau} - \eta^{\tau} e'(w^{\tau}) \tag{2.22}$$

目标函数 e 的数学性质通常是未知的，如果知道的话，就可以很简单地确定最小值。相反，给定数据 (X, Y)，我们只能对第 τ 次迭代的 w^{τ} 位置处的误差进行近似。事实上，就算给定所有数据样本的累积误差计算的梯度也只是真实误差函数梯度的近似，并不是真正的梯度。

很自然，如果我们得到的数据越多，梯度的近似值就会越接近真实值。如果我们有无限的数据，我们就可能生成真正的梯度。相反，数据越少，梯度近似就越差。图 2-10 显示了梯度下降算法的典型寻优路径。很显然总的移动方向指向碗的中心，就是我们要寻找的最优点。

公式 2.22 中的 η 通常被称为学习速率，η 的选择在学习算法中起着重要的作用。如果 η 太小，就会以缓慢的速度在参数空间中移动；如果 η 太大，那么一步的跨度就很大。这两种策略都有各自的优缺点。当确定是对梯度的较好近似时，采取较大的步长是合适的。但当不能确定前进方向是否正确时，就不能采取过大的步长。当我们使用小批量的数据或没有大的数据集时，近似的梯度是有噪声的。大的 η 还会遇到另一个问题，就是在局部极小值点附近来回跳跃而无法实际收敛。如图 2-10 所示，如果极小值点位于步长之内，那么就会跨越极小值，这就产生了在局部极小值点附近来回反弹的现象。如果 η 太小了，则需要大量的迭代才能到达极小值点。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

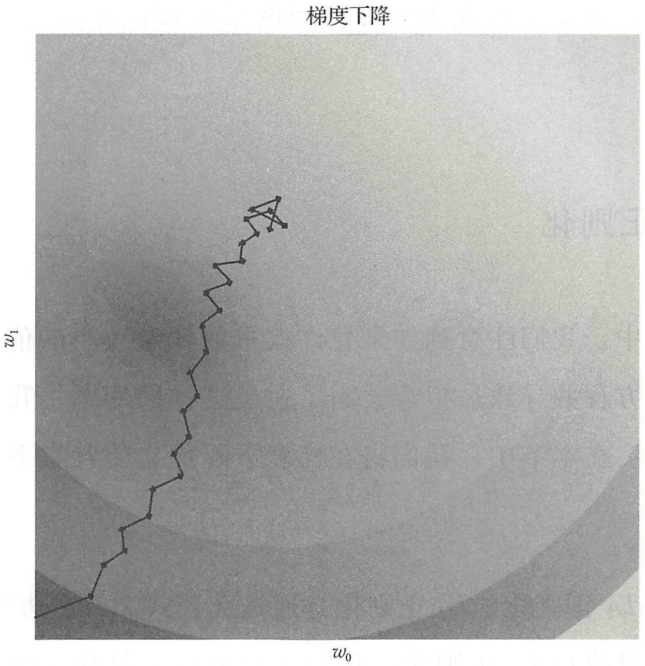


图 2-10 通过梯度下降算法到达中心的路径。我们从参数平面的顶部观察误差函数，环是等误差的等值线。黑线是每次迭代的位移，方形的点是每次迭代的参数组合

理想情况下，需要一个能动态变化的 η，因此迭代式 2.22 中的步长采用 η^{τ} 的形式，使之与 τ 有关。选择 η 的常用策略是在迭代早期阶段选择足够大的，这样就可以前进得更快，但经过一段时间的迭代后，让 η 变得更小，这样就容易收敛。通常，η 这种衰减过程被表达为迭代周期的函数，例如上面的 η^{τ} 通常表现为 τ 的分段线性或指数函数。

误差函数的二阶导数或 Hessian 矩阵（如果存在）会成为误差面曲率的指示器。使用 Hessian 矩阵的逆或泰勒展开近似是避免选择学习率的流行做法。空间越弯曲，我们应该采取越大的步长。当接近极小值时，函数变得更平坦，而在最小值时则完全平坦。综上所述，这些技术在机器学习

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

中得到了推广，梯度下降成为学习神经网络的最优算法之一（Rumelhart 等人，1986）。

2.9 几何正则化

在前一节中，我们注意到如果允许权重取任意大小的值，就可能得到不希望的回归方程并导致过拟合。为了避免这样的问题，我们简单讨论了正则化的概念。在本节中，我们将在梯度下降优化的背景下重新审视几何正则化。

首先考虑 L_2 正则化。L_2 正则化是通过在公式 2.8 右边增加 $\alpha_2 \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}$ 项来实现的。通过执行 L_2 正则化，我们把权重向量限制在以原点为圆心的圆内。很显然，$\boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}$ 是一个圆，高维情况下是个超球面。如果圆的半径增大，损失函数中的这个附加项就会增加。在一个特征空间中（如图 2-11 所示的二维空间），如果权重向量在圆之外，则损失函数会变大，因此要小心地限制其半径。在高维特征空间中，这一约束概念被延伸到超球面上，限制权重向量长度的增加。图 2-11 给出了这种约束的思想。

在上述约束条件下，最优解只能出现在圆周上。圆内部的任何点都会有更大的误差值，因为它们都会远离误差函数的谷底中心。这里需要在两种需求之间平衡。正则化项将尽可能地把权重向量推到靠近原点的位置，而损失函数则尽可能地将权重向量推到误差谷底。因此，最优解只能出现在正则化项的约束边界，本例的约束边界是圆周。

图 2-11 展示了这个性质，给定圆约束的唯一最优解显然是其圆周与

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

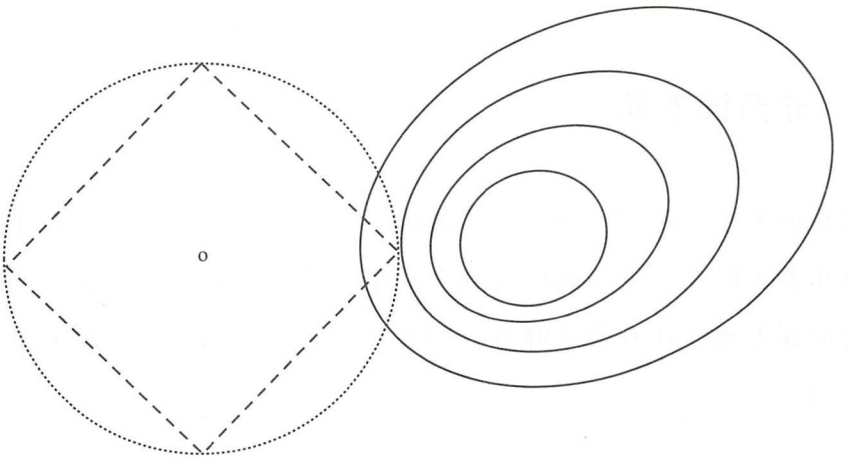


图 2-11 几何正则化。同心椭圆是误差函数的等值线。每条等值线是相同误差值的点的轨迹。虚线绘制的正方形是 L_1 约束，点线绘制的圆是 L_2 约束。横纵坐标轴分别代表参数 W_0 和 W_1，o 点表示空间原点

误差等值线相切的点。在这种情况下，误差函数仅在一点满足正则化，没有比接触点误差小的误差轮廓会出现在圆内。这限制了我们的搜索空间到圆周上的点。L_2 正则化规定，在权值的多项式组合中，交叉项不存在高概率。因为在寻找最优参数的过程中，我们把自己限制在了一个超球体的圆周上，所以我们将减少正在考虑的模型尺寸。

另外，L_1 正则化是一种更强大的正则化技术，可以得到稀疏权重。我们注意到，L_1 正则化对非零权重的惩罚比 L_2 要高得多，它限制权重值超过特定的值。L_1 正则化的几何形状是一个正方形，其重心在原点，如图 2-11 所示。虽然我们在这儿没有用数学证明它，但是 lasso（L_1）有很高的概率让误差函数和约束正方形在顶点处相交。由于约束正方形的顶点总是在坐标轴上，这意味着至少有一些权重将被强制为零。坐标轴上顶点对应的权值为零，从而实现权重的稀疏性，通常被称为权重衰减技术。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

2.10　非凸误差面

到目前为止，我们已经研究了碗状的凸误差面的梯度下降技术。凸误差面意味着可以在特征空间中找到全局最优解。一般而言，神经网络等复杂系统的误差面往往有多个峰谷。图 2-12 显示了具有多个高峰和低谷的典型误差面。

对于这类复杂的误差面，仍然可以使用以前的梯度下降技术来寻找最优解。但必须意识到，尽管存在许多这样的低谷，但一次只能到达其中的一个。因此可能需要运行多次梯度下降和交叉验证，才能找到一个足够好的局部最优解。寻优过程的终止点取决于初始化，如图 2-12 所示。

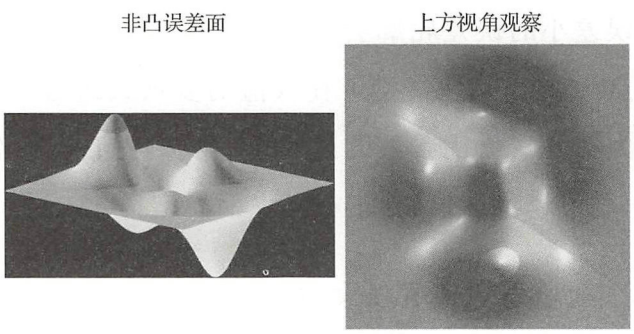


图 2-12　是一个较为复杂的、带有典型峰谷的非凸误差函数

由于在这些非凸表面上寻找最优解，它的初始条件很重要，最近提出了一些巧妙方法来初始化这种大型优化系统。我们将在接下来的章节中介绍其中的一些方法。最近的研究也表明，在大而深的神经网络和卷积神经网络（CNN）中，通常这些低谷都有相似的深度，这意味着可以使用其中

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

任何一个低谷作为最优点（Choromanska 等人，2015）。

2.11 随机梯度、批梯度及在线梯度下降

到目前为止，我们在公式 2.21 中使用了数据中的所有样本（X，y）来计算误差函数的梯度近似值，然后在公式 2.22 中更新权重。正如以前所看到的那样，一下子使用整个数据集有时是不太可能的，因此通常需要分批来处理。一种简单而极端的方法就是随机梯度下降法。记得我们的训练数据集有 n 个样本，可以使用逐个样本来做梯度估计。即在神经元上输入一个样本，然后检查它的输出和误差，并产生梯度近似。上述过程需要重写权重更新公式：

$$w^{n\tau+i} = w^{n\tau+i-1} - \eta^{n\tau+i-1} e'(w^{n\tau+i-1}) \tag{2.23}$$

同时

$$e'(w^{n\tau+i-1}) = -2x_i^{\mathrm{T}} y_i + 2x_i^{\mathrm{T}} x_i w^{n\tau+i-1} \tag{2.24}$$

在这里，$n\tau$ 次迭代被称为一个循环（epoch），一个循环是遍历整个数据集的迭代集合。在之前的公式 2.22 中，每个迭代都是一个循环，称为批梯度下降，因为一个迭代的梯度计算需要处理整批数据。

由公式 2.24 计算出的误差噪声将会非常大，而且极不稳定。为了减弱这一效应，我们大幅降低学习率，以更慢的速度在参数空间中移动。这样做是可行的，而且仍然能保证快速运行，因为一个样本就能更新一次权重，而不需要处理完所有样本才能更新。在批梯度下降过程中，我们对每

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

一个循环都可以自信地前进一大步，而在随机梯度下降中，我们可能前进了 n 个可疑的小步。

存在一个折中的做法：将数据集划分成小批量，并估计一个比纯粹随机好一点的梯度，再通过上面描述的过程学习。在现代的 CNN 和深度学习系统中，由于体系结构、内存和系统设计因素的影响，这种小批量随机梯度估计更可取。利用小批量随机梯度下降，我们可以有一个相当高的学习率，但搜索步数仍然很多。为 CNN 设计的现代的 GPU 使得小批量随机梯度下降比其他任何技术都要高效。

2.12　其他自适应学习率的更新规则

前面简单地提到过，避免选择学习速率和快速通过误差面的一种方法是使用 Hessian 矩阵的逆作为学习速率，这种方法常被称为牛顿法。考虑以下更新规则：

$$\boldsymbol{w}_i^{\tau+1} = \boldsymbol{w}_i^{\tau} - \boldsymbol{H}_i^{-1} e'(\boldsymbol{w}^{\tau}) \tag{2.25}$$

这里 $\boldsymbol{H}_i^{-1}$ 代表 Hessian 矩阵的逆矩阵的第 i 个权向量。使用 Hessian 矩阵要快很多，如果是一个凸碗状的误差函数，只需要一次迭代就能找到最优解。尽管它有很好的特性，但由于 Hessian 矩阵通常很大，超过大多数内存系统的容量，会导致牛顿法不可取。与参数数量相比，Hessian 矩阵具有指数级的规模，因此不适合有大量参数的深度学习系统。

存在一些其他的二阶更新方法，它们对内存需求与参数数量呈线性关

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

系。一种常用的方法称为 Adagrad，其更新规则如公式 2.26 所示：

$$w_i^{\tau+1} = w_i^{\tau} - \frac{\eta^{\tau}}{\sqrt[2]{\sum_{t=1}^{\tau} e'(w_i^t)^2 + \varepsilon}} e'(w_i^{\tau}) \tag{2.26}$$

可以把整个 $\frac{\eta^{\tau}}{\sqrt[2]{\sum_{t=1}^{\tau} e'(w_i^t)^2 + \varepsilon}}$ 看作是一个依赖于之前梯度的学习速率。通过在 t 上的求和，可以收集之前迭代中的累积二阶梯度。这个累积梯度包含了从过去到当前迭代梯度移动的平均方向。这样，Adagrad 可以减轻学习速率选择的负担（Duchi 等人，2011）。ε 是额外添加的一个假因数，目的是避免在梯度有缺陷或为零的情况下被零除。步长在很大程度上由先前迭代中产生的梯度大小控制。这种方法本质上使每个参数具有独立的学习速率，有助于更快的收敛。我们在每个独立维度中所采取的步长大小并不相同，由先前迭代在该维度中累积的梯度决定。需要注意的是，这里的分母是向量，梯度的每个维度都有一个值。

均方根传播（RMSPROP）是 Adagrad 的一种适应，它保持了梯度的移动平均值（Hinton，n. d）。RMSPROP 更新规则是：

$$w_i^{\tau+1} = w_i^{\tau} - \frac{\eta^{\tau}}{\sqrt[2]{m_{\tau}^i + \varepsilon}} e'(w_i^{\tau}) \tag{2.27}$$

同时

$$m_{\tau}^i = \rho m_{\tau-1}^i + (1-\rho) e'(w_i^{\tau}) \tag{2.28}$$

ρ 必须在 0 和 1 之间。这是一种类似于 Adagrad 的方式，由 Adam 进一步

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

发展而来（Kingma 和 Ba，2014）。

2.13　动量

如果我们相信梯度的近似值，就没有理由相信梯度会朝过于剧烈的方向移动。我们希望平滑参数空间中的移动路径。前一个小批量梯度估计在当前阶段仍然可能有用，因此可以把它想象成这个方向的“动量”（Polyak，1964）。这避免了由于小批量样本带入的噪声而导致的错误梯度近似，从而进一步避免在错误方向上的急剧移动。如果移动方向是连续的，动量会鼓励这个方向的移动；如果移动方向剧烈变化，动量会阻止这种变化。这种方法如果使用得当，可以加速动量最大的泛化方向上的学习，通常需要在共同方向上经过几次迭代的积累。

考虑用旧的平均速度 v 的衰减版本减去新梯度的加权版本，在一个方向上建立运动速度的变化量：

$$v_i^\tau = \alpha^\tau v_i^{\tau-1} - \eta^\tau e'(w_i^\tau) \tag{2.29}$$

现在我们已经用当前梯度的权重更新了速度，接下来就可以用这个速度间接地更新权重：

$$w_i^{\tau+1} = w_i^\tau + v_i^\tau \tag{2.30}$$

在这里，α 被设置成接近于 1 的值，因为我们相信速度是在以前循环上建立起来的，但从 0.5 开始会比较好，这样可以减少早期迭代的影响，因为刚开始的学习通常是混乱的。α 表示我们对速度的信任度。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

另一种常用的动量技术是 Nesterov 提出的加速梯度（NAG）方法，它先用速度来更新权值，到后面才开始用累积速度（Nesterov，1983）。NAG 的更新步骤是

$$w_i^{\tau+1} = w_i^{\tau} + \alpha^{\tau} v_i^{\tau} - \eta^{\tau} e'(w_i^{\tau}) \tag{2.31}$$

动量和二阶梯度更新极大地改变了学习在深度神经网络和卷积神经网络中的收敛速度和稳定性。在引用文献中讨论了动量的重要性，对想深入研究的读者来说是个很有吸引力的话题（Sutskever 等人，2013）。

本章小结

在本章中，我们研究了线性回归和连续值标签的建模。从有监督机器学习开始，创建了一个简单的监督学习数据集。为了使问题变得简单，我们对数据集做了线性假设，在线性假设下建立线性模型并将其参数化。我们用最小二乘法得到了参数的解析解，并用极大似然估计解释了最小二乘解。

我们对模型的参数施加多种约束，研究了正则化和岭回归模型。正则化使我们的学习变得稳定，避免了权重的爆炸。附加的约束条件加入到我们已经存在的解析解模型中，会发现正则化回归更容易求解。

我们还研究了这些线性模型可能产生的一些潜在问题。如果数据集不是线性的，那么我们的假设就会不成立，必须寻求一个非线性的解。利用基函数的扩展，可将特征空间映射到一个非线性空间，在这个非线性空间

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

中求解线性回归问题。这使我们能够用线性回归技术来解决非线性回归问题。同时，我们也研究了过拟合的问题，并将正则化作为一种减弱过拟合的方法。我们研究了L_1和L_2范数及其对模型光滑性的影响。我们也研究了如何使用交叉验证来更好地评价模型和平均化模型。

并非所有的回归问题都有解析解。当数据集很大，解析解存在不可逆矩阵，即使存在也可能在实际上行不通。在本章中，我们还扩展了一开始对线性和非线性回归的解析解的认识。我们使用优化技术，特别是梯度下降法来解决问题并找到最优的权重。为了更小批量地使用我们的数据，我们还研究了随机梯度下降法的应用，研究了优化过程中不同学习速率的影响。根据优化任务的不同阶段，使用自适应的学习速率可以更稳定地到达局部最小值。然后我们继续学习了 Adagrad、牛顿法和其他二阶学习技巧。我们还研究了正则化的几何结构以及不同的正则化器如何影响我们在优化时让权重移动的区域。另一种控制和移动权重的方法是使用动量，我们研究了各种形式的动量包括 Polyak 和 Nesterov 提出的动量技术。

参考文献

Boyd, Stephen and Vandenberghe, Lieven. 2004. *Convex Optimization*. Cambridge: Cambridge University Press.

Button, Katherine S, Ioannidis, John PA, Mokrysz, Claire et al. 2013. Power failure: Why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience* 14: 365–376.

Choromanska, Anna, Henaff, Mikael, Mathieu, Michael et al. 2015. The loss surfaces of multilayer. *AISTATS*.

Duchi, John, Hazan, Elad, Singer, Yoram. 2011, July. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of*

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Machine Learning Research 12: 2121–2159.

Efron, Bradley and Gong, Gail. 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician* 37: 36–48.

Felzenszwalb, Pedro F, Girshick, Ross B, and McAllester, David. 2010. Cascade object detection with deformable part models. *2010 Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2241–2248). IEEE.

Galton, Francis. 1894. *Natural Inheritance* (Vol. 5). New York, NY: Macmillan and Company.

Golub, Gene H, Heath, Michael, and Wahba, Grace. 1979. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* 21: 215–223.

Gunn, Steve R. 1998. *Support vector machines for classification and regression.* ISIS technical report.

Hinton, Geoffrey, Srivastava, Nitish, and Swersky, Kevin. n.d. *Overview of mini-batch gradient descent.* Retrieved from https://www.coursera.org/learn/neural-networks

Hoerl, Arthur E and Kennard, Robert W. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12: 55–67.

Jefferys, William H and Berger, James O. 1992. Ockham's razor and Bayesian analysis. *American Scientist* 80: 64–72.

Kingma, Diederik and Ba, Jimmy. 2014. *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980.

Kohavi, Ron. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1137–1145.

Lawson, Charles L and Hanson, Richard J. 1995. *Solving Least Squares Problems* (Vol. 15). Philadelphia, PA: SIAM.

Le Cessie, Saskia and Van Houwelingen, Johannes C. 1992. Ridge estimators in logistic regression. *Applied Statistics* 41(1): 191–201.

Marquardt, Donald W and Snee, Ronald D. 1975. Ridge regression in practice. *American Statistician* 29(1): 3–20.

Nadaraya, Elizbar A. 1964. On estimating regression. *Theory of Probability and Its Applications* 9: 141–142.

Nesterov, Yuri. 1983. A method of solving a convex programming problem with convergence rate O (1/k2). *Soviet Mathematics Doklady* 27: 372–376.

Neter, John, Kutner, Michael H, Nachtsheim, Christopher J et al. 1996. *Applied Linear Statistical Models* (Vol. 4). Pennsylvania Plaza, NY: McGraw-Hill/Irwin.

Polyak, Boris Teodorovich. 1964. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 4: 1–17.

Ramsay, James O. 2006. *Functional Data Analysis.* Hoboken, NJ: Wiley Online Library.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. 1986. Learning representations by back-propagating errors. *Nature* 323: 533–538.

Seber, George AF and Lee, Alan J. 2012. *Linear Regression Analysis* (Vol. 936). Hoboken, NJ: John Wiley & Sons.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Shao, Jun. 1993. Linear model selection by cross-validation. *Journal of the American Statistical Association* 88: 486–494.

Shawe-Taylor, John and Cristianini, Nello. 2004. *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge University Press.

Sutskever, Ilya, Martens, James, Dahl, George E et al. 2013. On the importance of initialization and momentum in deep learning. *International Conference on Machine Learning*, Vol. 28, pp. 1139–1147, Atlanta, Georgia.

Vapnik, Vladimir, Golowich, Steven E, Smola, Alex et al. 1997. Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems*, pp. 281–287.

Zhang, Ping. 1993. Model selection via multifold cross validation. *Annals of Statistics* 21: 299–313.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

第3章
Convolutional Neural Networks in Visual Computing

人工神经网络

前面章节中，介绍了给定数据集下几种对多维向量分类和回归的方法。本章将介绍人工神经网络（Artificial Neural Networks，ANN），目前一些最先进的深度学习算法均以此为基础。尽管某些 ANN 文献受生物系统的启发，甚至已用硬件实现，如 McCulloch-Pitts 神经元（McCulloch 等人，1943），但大部分网络已设计为与生物神经网络无直接关联的简单程序。从实践的角度出发，本章主要讨论 ANN 的算法。

考虑第 2 章的回归问题：数据集 $D=\{(\boldsymbol{x}_i,y_i),i\in[1,2,\cdots,n]\}$，其中 x_i 表示 d 维数据，y_i 为类标签，n 为数据集大小。图 3-1 将线性回归问题描述为线性神经元的前馈网络（feedforward network），其中圆圈为处理单元；由于线性回归的处理单元都为线性，因此第一层的每个单元均是恒等函数。输出是与 y_i 连接的单元，其值由输入与对应边上的权重线性组合的和得到。虽然这只是上述问题的另一解决方案，但它在处理更复杂问题时却表现出极好的特性。

如图 3-1 所示，术语“前馈”是指处理单元只能由输入到输出方向进行运算。本章我们主要考虑前馈网络结构，但值得注意的是，在数据处理流程中，网络可以包含反馈路径，而不仅仅局限于前馈路径。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

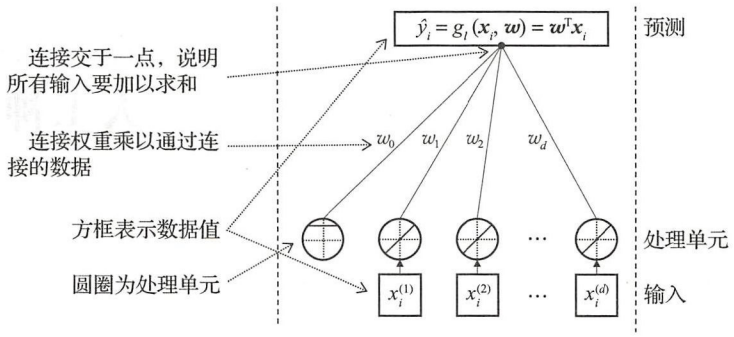


图 3-1　线性回归的前馈网络示意图

3.1　感知器

第 2 章主要讨论了回归与优化，如何对“输入-输出”关系进行建模和求解最优模型参数。视觉计算应用的大部分任务是执行有监督分类，比如通过支持向量机（Gunn，1998；Vapnik 等人，1997）。这些任务都能看作图 3-1 的变体。例如，对于一个二分类问题，所选的数据集 D 中，标签 y_i 可取两个值，即 0 或 1。尽管可以有其他的可能取值，这里假定用 0 或 1 作为 y_i 的两个取值。由于未考虑标签的二元属性就直接进行线性回归，学到的模型可能在预测中效果不会很好。一种自然的修正方式就是通过一个阈值函数输出，即将输出表示为：

$$g_P(\boldsymbol{x}_i, \boldsymbol{w}) = \begin{cases} 1, & (\boldsymbol{w}^T \boldsymbol{x}_i > 0) \\ 0, & \text{其他} \end{cases} \tag{3.1}$$

简便起见，阈值设为 $\boldsymbol{w}^T \boldsymbol{x}_i = 0$。阈值的设置有多种方法，后面会详细介绍。由于阈值函数的输出指示该神经元是否被激活，因此也称为激活函数（activation function）。在前一节的例子中不存在激活函数，或者说激活函

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

数中仅仅是个恒等式。由于激活函数仅仅起到门限的作用，因此它可以独立定义为：

$$a_P(t) = \begin{cases} 1, & (t > 0) \\ 0, & \text{其他} \end{cases} \tag{3.2}$$

本例中，t 为线性回归的输出。在此之上，我们建立了一个激活（阈值）函数。

上述神经元本质上为感知器或线性阈值神经元，该模型和 Rosenblatt 提出的使用 McCulloch-Pitts 神经元实现分类（Rosenblatt，1958）的模型相似，如图 3-2 所示。在 20 世纪 60 年代，人们对该感知器、它的变种及其特征空间中的几何结构进行了详尽研究，这也是人工神经网络的第一波浪潮，在其局限性被发现后不久，人工神经网络的这波势头就变得平缓了（Minski 等人，1969）。

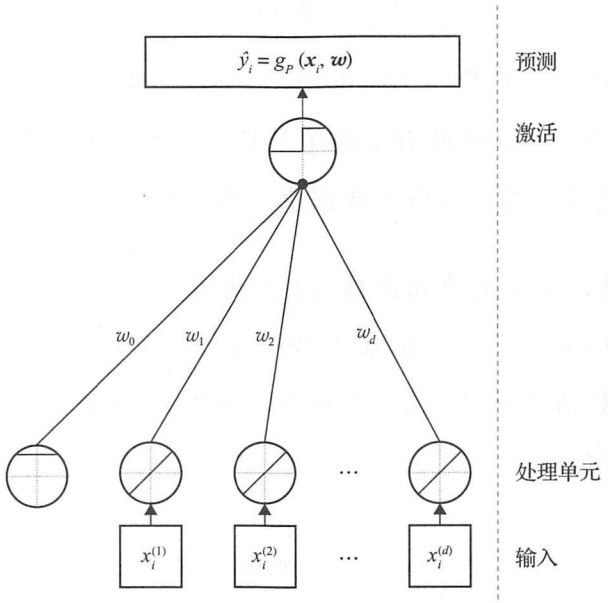


图 3-2 Rosenblatt 感知器

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

最开始用于训练网络的算法称作感知器算法。它是个有效的算法，因为它学到的感知器权重能保证得到一个较好的决策边界。它同时也是一个在线学习算法，可以进行逐样本学习。图 3-3 中给出了 Rosenblatt 的在线感知器算法。

Input

Training set $D = \{(\boldsymbol{x}_i, y_i), i \in [1,2,...n]\}$, $y_i = [0,1]$.

Initialization

Step 1: Initialize the weights and threshold for the network.
Step 2: Weights may be set to 0 or small random values.

Iterate

for $t \rightarrow$ convergence
{
　for each sample $\boldsymbol{x}_i$ with label y_i:
　{
　　Step 1: Compute the output $\hat{y}_i$ of the network.
　　Step 2: Estimate the error of the network $y_i - \hat{y}_i$.
　　Step 3: Update the weight $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + e(\boldsymbol{w})\boldsymbol{x}_i$.
　}
}

图 3-3　感知器算法

从 20 世纪 60 年代（Novikoff，1962）到最近几年（Mohri 和 Rostamizadeh，2013），这些看似简单的算法的收敛性/性能已经有过证明和研究，特别是针对线性可分的数据集。感兴趣的读者可参考上述文献。

有意思的是，在原始感知器被提出的同时，David Cox 于 1958 年提出 Logistic 回归（Cox，1958）。在基本的 Logistic 回归中，关键任务是通过 Logistic 回归函数估计 y 取 0 或 1 的概率。使用 g_σ 函数的模型 $P(y=1|\boldsymbol{x}) = g_\sigma\{\boldsymbol{x}, \boldsymbol{w}\}$ 定义为：

$$P(y=1|\boldsymbol{x}) = g_\sigma\{\boldsymbol{x}, \boldsymbol{w}\} = \frac{1}{1+e^{-\boldsymbol{w}^{\top}\boldsymbol{x}}} \tag{3.3}$$

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

其中激活函数为：

$$a_\sigma(t) = \frac{1}{1+e^{-t}} \tag{3.4}$$

模型的任务为在给定数据集下找到 w 的极大似然解。

极大似然估计通常不存在解析解，因此需要通过数值优化方法，如使用第 2 章中的迭代优化方法更新 w，目标函数为其对数似然。由于第 2 章已有对梯度下降的讨论，因此这里不赘述具体过程。感兴趣的读者可查阅相关文献获得信息（Hosmer Jr 等人，2004）。

$g_\sigma(\cdot)$ 和 $g_P(\cdot)$ 具有相似性，首先它们都是激活函数。如图 3-4 所示，$g_\sigma(\cdot)$ 可看作是一种软阈值函数，可以用它替代图 3-2 中感知器的硬阈值单元。如图 3-5 所示，Logistic 函数是连续且可微的。可微是允许我们建立误差梯度的一个重要特性，如第 2 章中的线性回归所看到的那样。因此，第 2 章中通过迭代搜索 w 的梯度法和其他优化技术都能在这里使用。

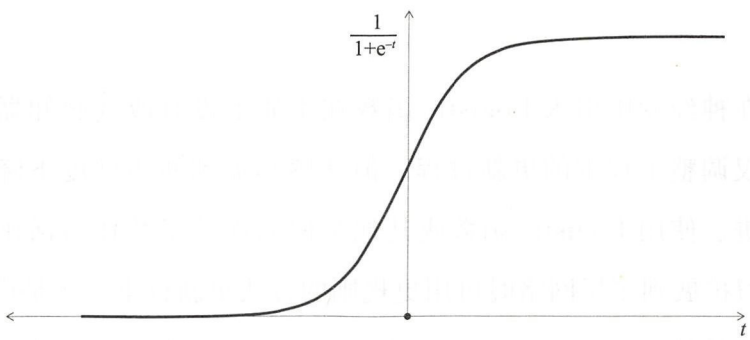


图 3-4 标准逻辑函数

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

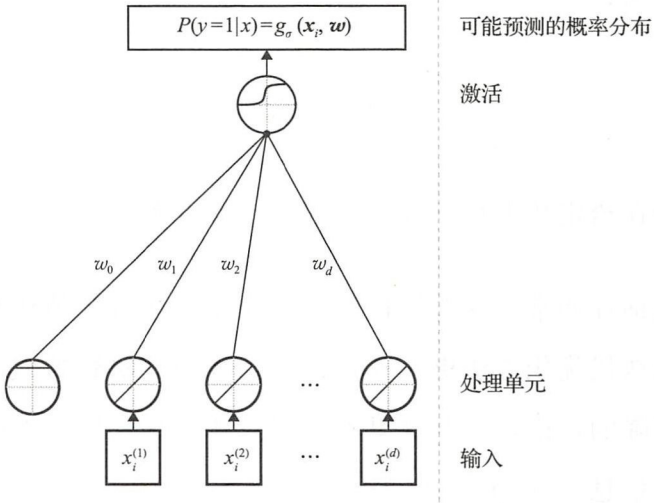


图 3-5　逻辑神经元

与线性回归一样，定义损失函数 $e(w)$，然后使用梯度下降方法更新权重：

$$w^{\tau+1} = w^{\tau} - \eta^{\tau} e'(w^{\tau}) \tag{3.5}$$

其中 η^{τ} 为标量，表示用来控制每一轮迭代中权重调整幅度，又称为学习速率。

尽管在神经元中引入 Logistic 函数在本质上没有改变感知器算法的基本思想，仅调整了权重的更新过程。但从感知器到使用梯度下降在计算上是重大改进。使用 Logistic 函数或其他类似的连续函数作激活函数的好处是，当学习扩展到多层网络时可用更规则的方法更新权重，于是产生了反向传播（BP）算法（Bryson，1961；Kelley，1960）。在所有深度学习和神经网络的研究中，反向传播算法都起着重要作用。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

通常，学习的目的是最小化已定义好的损失函数，现在我们可以定义二分类以外的其他形式的损失函数。实际应用中会定义得更灵活，最新使用各种损失函数的深度学习应用将在后续章节中介绍。

图 3-5 的神经元模型是扩展为多输出节点模型，如图 3-6 所示，这在处理多于两个类别的问题中很有必要。如果有 C 个类别的分类，那么我们需有 C 个输出节点。

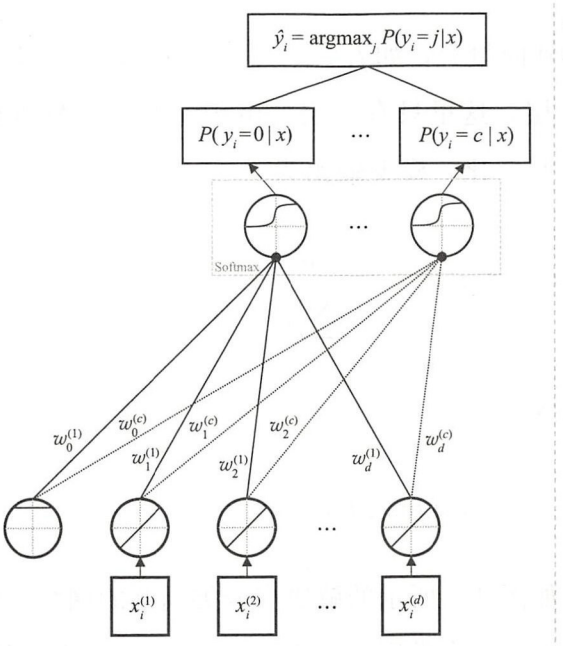


图 3-6 输出多于一个节点的网络

这种网络输出的不是类标签的直接预测，而是类的概率分布，如果 $\hat{y}$ 是预测标签，那么它就是上述输出概率中的最大者。假设有 C 个类，我们会得到维度为 C 的输出向量。softmax 层之前的点积表示为：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$$\boldsymbol{l}_1 = \boldsymbol{W}^{\mathrm{T}} \cdot \boldsymbol{x}_i \tag{3.6}$$

其中 $\boldsymbol{W}$ 是 $d \times c$ 的矩阵：

$$\boldsymbol{W} = \begin{bmatrix} w_0^{(1)} & w_0^{(2)} & \cdots & w_0^{(c)} \\ w_1^{(1)} & w_1^{(2)} & \cdots & w_1^{(c)} \\ \vdots & \vdots & \ddots & \vdots \\ w_d^{(1)} & w_d^{(2)} & \cdots & w_d^{(c)} \end{bmatrix} \tag{3.7}$$

通过公式 3.6，得到向量 l_1，如 $l_1=[\,l_1^{(1)},l_1^{(2)},\cdots,l_1^{(c)}]$，其中 c 为类的个数。下标 1 表示第一层，这里只有一层。由于这些不是标准化向量，这里选择 softmax 函数作归一化，模型输出为：

$$P(y_i = j \mid x_i) = \sigma(l_1^{(j)}) = \frac{e^{l_1^{(j)}}}{\sum_{k=1}^{c} e^{l_1^{(k)}}},\ j \in [1,2,\cdots,c] \tag{3.8}$$

预测类别为概率最大的类：

$$\hat{y}_i = \mathrm{argmax}_j P(y_j = j \mid x_i) \tag{3.9}$$

因此，我们得到了如图 3-6 所示的解决多分类问题的网络。建立与线性回归一样的对数似然，由于每个类别 $j \in [1,2,\cdots,c]$ 对应概率 $P(y_i=j \mid x)$，对数似然表示为：

$$l(\boldsymbol{W}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \mathbb{I}\, j(y_i) \log(P(y_i = j \mid x_i)) \tag{3.10}$$

其中，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$$\mathbb{I}_j(y_i) = \begin{cases} 1, & (y_i = j) \\ 0, & \text{其他} \end{cases} \tag{3.11}$$

为指示函数。指示函数仅在一个类中取 1，其他类取 0。这意味着最大化对数似然时，本质上是最大化正确类的概率。现在，我们对负 $l(w)$ 作梯度下降，即 logistic 的负对数似然。实现梯度下降的程序可通过简单扩展第 2 章中的算法得到，留给读者思考。在后面介绍反向传播算法时我们会详细介绍。注意，公式 3.10 得到的为整个数据集的平均似然，这种平均度量允许我们使用小批量数据（mini-batches）的平均似然进行类似的学习过程。同时，这也使得我们可以使用相同的学习速率而不用考虑小批量数据的大小。由于似然本身已用第 n 批次的样本大小来标准化，因此，即使使用随机梯度下降，学习速率也不依赖批量数据的大小。

我们已经研究了神经元的不同类型，包括感知器、线性回归神经元和 softmax 感知器。不同问题可以尝试用不同类型的神经元去解决。例如，使用最大边缘神经元生成最大边缘，或通过以下似然函数生成支持向量机（SVM）类型的分类器：

$$l(\boldsymbol{W}) = \max(0, 1 - y \cdot \hat{y}) \tag{3.12}$$

上述似然函数的使用需要将标记从 [0,1] 转换为 [−1,1]。如果预测与真实值匹配，似然值接近 0；如果预测错误，似然值会较大。实际中有多种不同类型的神经元，这里只研究那些流行的，与计算机视觉相关的部分。读者可通过参考文献来了解和研究各种神经元。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

3.2　多层神经网络

感知器算法和 Logistic 回归会生成线性决策边界，因此，若两个类线性可分，则可以解决。然而，单层感知器无法处理非线性分类问题，图 3-7 的“异或（XOR）”问题就是一个例子。如前面章节所述，使用基函数展开并使用高阶转换可以解决该问题（Powell，1977）。这就引出了选择什么样的基函数空间来映射特征才能在该空间中实现线性可分的问题。

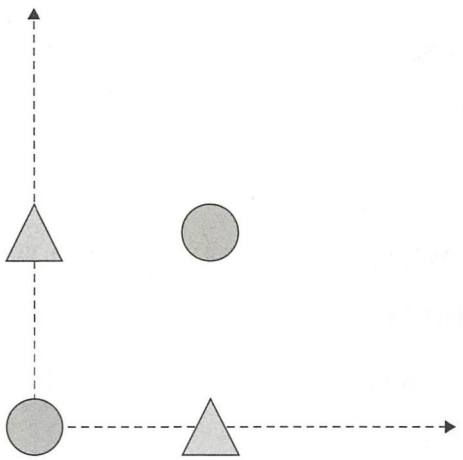


图 3-7　二维特征空间的异或问题。三角形和圆表示两种类别

解决该问题的一种方法是使用多层神经网络（MLNN），使用 MLNN 可学习到这种函数变换。计算机视觉领域的科学家称其为特征提取。第 1 章我们介绍了特征提取，并讨论了一些计算机视觉中受欢迎的人工特征。从某种意义上说，这些特征是原始像素的非线性变换，它由领域专家设计和调整而来。原始像素转换到特征空间后可以更有效地完成分类任务。实

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

践中，由于这些转换是手工设计的固定过程，因此通常不适用于使用不同损失函数的分类问题。

多层神经网络可以学习这样的转换。更深的网络能学习到更复杂的特征空间，在这些特征空间中可完成线性分类的任务。

基本感知器的处理单元是线性的，那么，非线性的处理单元或许具有生成非线性决策边界的能力。引入非线性的处理单元有多种形式，例如通过引入非线性径向基函数来实现径向基函数网络（Lowe 和 Broomhead，1988）。另一种方法是简单地在基本感知器中插入至少一层权重可调的非线性单元，就得到了 MLNN，由于它仍然使用传统的感知器连接，这样的网络也称为多层感知器（MLP）。由于这两种术语的滥用，我们将其作为同一含义使用。图 3-8 为一个简单的多层神经网络，由于中间层在输入 x

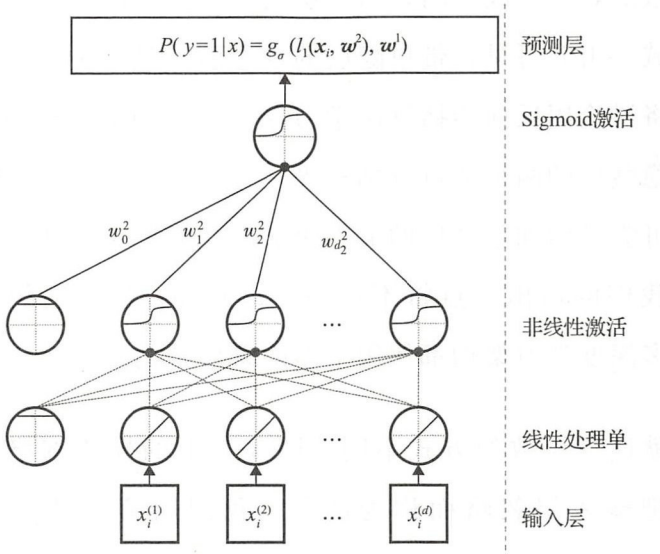


图 3-8 简单的多层神经网络

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

和输出 y 之间隐藏，因此称为隐藏层。隐藏层有 d_2 个节点，而输入层的节点个数由输入本身的维数决定，即 d 个。

这种简单的感知器组合可以解决图 3-7 的异或问题，同时也学到了有效的特征映射。显然，图 3-8 的分类器 g 是在 l_1 层的变换基础上构建的，其中 l_1 层又是由输入层变换得到的。同样，l_1 层的这种变换依赖参数 w^2。因此，可以直观地看到单隐藏层网络内部如何映射到新的特征空间。由于隐藏层上的节点对信号求和后进行了非线性激活，因此也是 Logistic 回归神经元。模型输出之前已有一个由多个神经元组成的隐藏层。

讨论如何学习多层神经网络前，我们先给出一个解决异或问题的方法，如图 3-9 所示，隐藏层由两个节点和一个偏置组成。在隐藏层中，仅当两个输入均为 1，$\wedge$ 节点才会激活，即“与”门。当任意一个输入为 1，$\vee$ 节点就会激活，即“或”门；第 2 层为异或门，图 3-9 中给出了特征空间和分类区域。由此可见，带单隐层的神经网络是一种解决异或问题的方法。下一节将讨论用反向传播算法学习这个神经网络的权重。给定神经网络的结构，隐藏层的两个节点分别充当“与”门、“或”门来模拟“异或”功能。读者可尝试增加隐藏层的节点个数并比较结果。类似地，处理多分类问题时，我们的输出节点将不止一个；此外，隐藏层的数量也可能更多，目前许多深度学习架构都使用了多隐藏层结构。

不同文献计算层数的方式不同，图 3-9 中的异或网络，一种称为三层，另一种把输入层忽略而称为两层。我们用第一种方式计算，使用“$n_{\mathrm{I}}\rightarrow n_{\mathrm{H}}\rightarrow n_{\mathrm{C}}$　MLP”来表示三层网络，有 n_I 个输入节点，第一个隐藏层有 n_H 个隐藏节点，有 n_C 个输出节点。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

以上述三层的多层感知器为例，我们给出 $n_I \rightarrow n_H \rightarrow n_C$ 前馈网络输入与输出的数学关系：

$$\hat{y} = a_C(a_H(a_I(\boldsymbol{x}), \boldsymbol{w}_H), \boldsymbol{w}_C) \tag{3.13}$$

我们在输入层、隐藏层和输出层分别使用了 a_I、a_H 和 a_C 三种激活函数，说明允许各层使用不同的激活函数，甚至包括输入层。事实上，每个隐藏或输出节点也可以使用各自的激活函数；为了实践方便，我们习惯对相同层使用相同的激活函数。

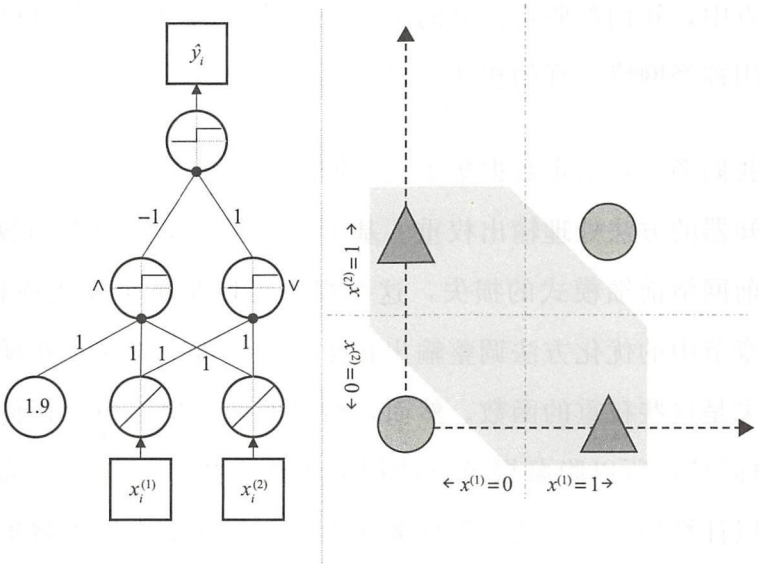


图 3-9 MLNN 解决二维空间中的 XOR 问题

尽管上述形式很简单，但这种三层网络的输入-输出表示具有非常好的通用性：理论上它能模拟从输入到输出的任何连续函数，前提是拥有足够多的隐藏单元 n_H、适当的非线性变换及定义适当的权重。这通常认为由 Kolmogorov（1956）的定理证明得到，然而，在多层网络相关性上还是受

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

到了实用性方面的争议（GiROSI 和 Pojo，1989）。

抛开严格证明，直观上我们仍能理解三层网络为何能够用于映射任何输入-输出的关系。例如，从第 1 章中傅里叶分析的角度来看，任何连续函数都能由无穷次调和函数（或傅里叶基）以任意精度逼近，并考虑隐藏节点可以模拟函数，从而利用隐藏节点形成函数的期望近似。此外，利用图 3-1 至图 3-4 中回归与感知器的简单关系，由于隐藏层可以作为输入，投影到更低维空间来完成分类（或回归）（Friedman 和 Stuetzle，1981；Lingjaerde 和 Liestol，1998），将多层感知器与投影追踪回归联系起来。在两个论点中，我们都假定网络的学习是寻找好的“特征”，以便于决策，如我们使用神经网络实现的想法一样。

现在我们考虑在给定数据集下训练网络参数的问题。显然，可以用训练单层感知器的方法处理输出权重。基于任何预定义的损失函数，我们首先计算当前网络前馈模式的损失，这个损失可以是单个输入或批量输入，并用之前章节中的优化方法调整输出权重，如梯度下降法。在梯度下降方法中，损失是这些权重的函数。然而，给定的训练集不包含任何隐藏节点的“目标标记”，所以隐藏层节点的损失无法直接计算。由于输出层有目标标记可以计算损失，因此关键任务变为了如何在输出层中将损失适当地分配到每个隐藏节点。

3.3　反向传播算法

反向传播（BP）算法是实现多层感知器参数学习的主要方法。文献中

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

的 BP 算法是逐渐改进的，这些贡献并不归功于个人或某篇文章。尽管如此，20 世纪 80 年代的一篇文章唤起了人们对该算法的关注（Rumelhart，1988）。基于梯度下降学习的 BP 算法思想非常简单，它仅依靠微分的链式法则在输出层和任何隐藏节点间构建连接。该连接将网络的损失反向传回网络的任何层或节点，以便在减少损失的方向进行调整。图 3-10 的网络说明了 BP 算法。

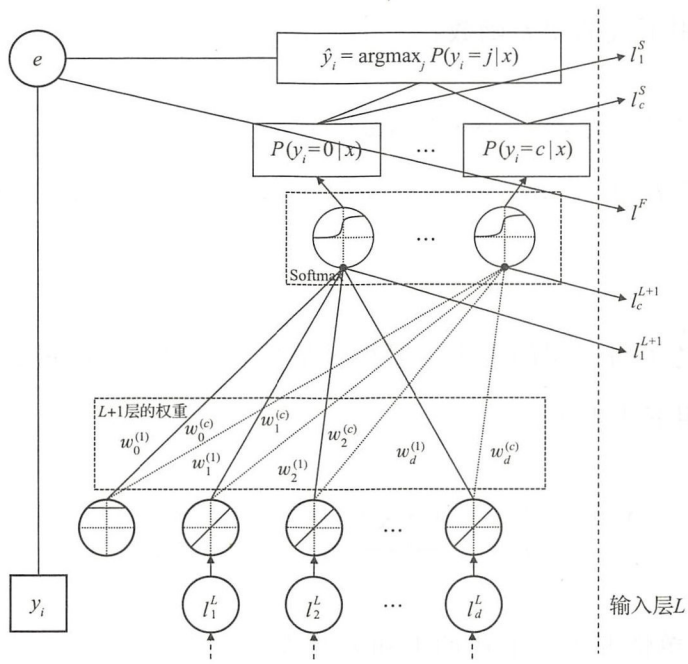


图 3-10 前馈网络用于反向传播网络（展示了最后一层中第 i 个样本的 softmax 层和误差层）

为能够对任何网络参数 w 使用梯度下降，我们需找到误差 e 关于该参数的梯度。即计算 $\partial e / \partial w_{i,j}^{(k)}$，$\forall w_{i,j}^{(k)}$ 表示网络第 i 层的第 j 个神经元的第 k 维。我们将忽略第 i 层的下标，而仅考虑一个参数的层来展示使用

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

链式法则隔层导出导数，这能扩展到其他层。l^F 为最后层的输出，它在公式 3.10 中已经定义。l_j^S 在公式 3.8 中定义，表示 softmax 层的输出。l_j^{L+1} 是由 $l_j^{L+1}=\sum_i w_i^{(j)} l_i^{(L)}$ 定义的点积输出的，式中忽略了权重符号的层数，其中 l_j^L 为上一层的输入。我们希望求得 $\partial e/\partial w_j$ 项，一旦我们求得某个 j 时的导数，就能依次对所有 j 做相同操作；一旦我们能对图中位置求解，就能进行扩展，为所有层求解。

考虑使用下式的误差函数：

$$e(w)=l^F\left[\begin{array}{l} l_1^S\{l_1^{L+1}(\langle l_1^{(L)},w_1^{(1)}\rangle,\cdots\langle l_d^{(L)},w_d^{(1)}\rangle)\},\cdots\{l_c^{L+1}(\langle l_1^{(L)},w_1^{(c)}\rangle,\cdots\langle l_d^{(L)},w_d^{(c)}\rangle)\}, \\ \cdots l_c^S\{l_1^{L+1}(\langle l_1^{(L)},w_1^{(1)}\rangle,\cdots\langle l_d^{(L)},w_d^{(1)}\rangle)\},\cdots\{l_c^{L+1}(\langle l_1^{(L)},w_1^{(c)}\rangle,\cdots\langle l_d^{(L)},w_d^{(c)}\rangle)\} \end{array}\right] \tag{3.14}$$

嵌套误差描述了每一层的上下关联性。显然，任何权重向量 w_k，其偏导数 $\partial e/\partial w_k$ 可用下式表示：

$$\frac{\partial e}{\partial w_k}=\sum_{i=1}^{c}\sum_{j=1}^{c}\frac{\partial l^F}{\partial l_i^S}\frac{\partial l_i^S}{\partial l_j^{L+1}}\frac{\partial l_j^{L+1}}{\partial w_k} \tag{3.15}$$

在 $c=2$ 的简单情况下，上述的求和展开为：

$$\begin{aligned}\frac{\partial e}{\partial w_k}=&\frac{\partial l^F}{\partial l_1^S}\frac{\partial l_1^S}{\partial l_1^{L+1}}\frac{\partial l_1^{L+1}}{\partial w_k}+\frac{\partial l^F}{\partial l_1^S}\frac{\partial l_1^S}{\partial l_2^{L+1}}\frac{\partial l_2^{L+1}}{\partial w_k}\\&+\frac{\partial l^F}{\partial l_2^S}\frac{\partial l_2^S}{\partial l_1^{L+1}}\frac{\partial l_1^{L+1}}{\partial w_k}+\frac{\partial l^F}{\partial l_2^S}\frac{\partial l_2^S}{\partial l_2^{L+1}}\frac{\partial l^{L+1}}{\partial w_k}\end{aligned} \tag{3.16}$$

显然，求和式中出现了重复项，这些为不必要计算，我们将其重写为

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

一种高效计算梯度的公式：

$$\frac{\partial e}{\partial w_k}=\frac{\partial l_1^{L+1}}{\partial w_k}\left[\frac{\partial l^F}{\partial l_1^S}\frac{\partial l_1^S}{\partial l_1^{L+1}}+\frac{\partial l^F}{\partial l_2^S}\frac{\partial l_2^S}{\partial l_1^{L+1}}\right]+\frac{\partial l_2^{L+1}}{\partial w_k}\left[\frac{\partial l^F}{\partial l_1^S}\frac{\partial l_1^S}{\partial l_2^{L+1}}+\frac{\partial l^F}{\partial l_2^S}\frac{\partial l_2^S}{\partial l_2^{L+1}}\right] \tag{3.17}$$

或者，

$$\frac{\partial e}{\partial w_k}=\frac{\partial l_1^{L+1}}{\partial w_k}[\Delta(S_1)]+\frac{\partial l_2^{L+1}}{\partial w_k}[\Delta(S_2)] \tag{3.18}$$

向量 $\Delta(S)$ 包含 $\Delta(S_1)$ 和 $\Delta(S_2)$，由于 S 层中仅需知道 $L+1$ 层的输出，所以不依赖 $L+1$ 层；这里可将其作为从顶层到 $L+1$ 层的输入。

考虑带参数 w_q 的 q 层，其输入向量为 l^{q-1}，输出向量为 l^q，它的内部会计算梯度，即 $\partial e/\partial w_q$。这里我们仅需从之前层得到梯度 $\Delta(q)$。得出梯度后，将梯度 $\Delta(q-1)$ 传播到下一层。

这就是 BP 算法，自输入层向输出层前馈地计算误差，再从输出层到输入层逐层传播梯度，直至所有层都计算出梯度误差。当我们得到所有权重的误差梯度，就可以使用之前的优化方法实现学习。

3.4 改进的反向传播算法

经过发展，产生了大量改进多层神经网络的方法，包括理论性更强及更具实用性的改进措施。第 2 章中研究了一些优化的内容，如动量、学习

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

率修正和二阶方法。下面介绍一些广泛使用的方法，将在后文详细阐述。

3.4.1　激活函数

上文只讨论了阈值、恒等和 softmax 激活函数，下面讨论几个最新文献中使用的激活函数。

经过充分论证的激活函数包括平方函数和双曲正切函数。在实用学习和稳定性相关问题中，上述函数还未证明其有效性。最常用的现代激活函数之一是整流器，或称 ReLU。ReLU 引入于 20 世纪，并于 2010 年取得巨大成功（Hahnloser，2000；Nair 和 Hinton，2010）。ReLU 的形式如下：

$$a_{\mathrm{ReLU}}(t) = \max(0,t) \tag{3.19}$$

这是个不光滑函数。实现时采用以下 ReLU 形式：

$$a'_{\mathrm{ReLU}}(t) = \ln(1+e^{x}) \tag{3.20}$$

这是 ReLU 的光滑函数表示，称为 softplus。

ReLU 激活函数如图 3-11 所示，其中 ReLU 更陡，学习更快（He 等人，2015；Nair 和 Hinton，2010）。

激活函数添加噪声可扩展为带噪声的 ReLU，定义如下：

$$a_{\mathrm{ReLU}}(t) = \max(0,t+\varphi),\ \text{with}\ \varphi \sim \mathcal{N}(0,\sigma(t)) \tag{3.21}$$

带噪声的 ReLU 函数会在激活时产生些随机误差，这允许网络在学习时出现少量错误。考虑到误差表面凹凸不平，这有助于探测到通过严格梯度下

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

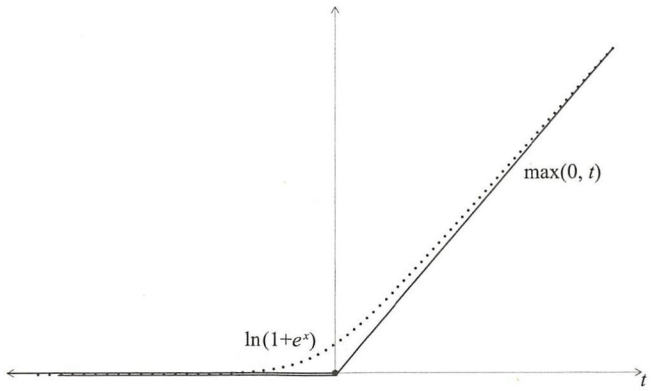


图 3-11 ReLU 激活函数及其解析近似

降无法达到的位置；同时，这也能缓解过拟合。带噪声的 ReLU 在某些计算机视觉任务中表现出良好性能。

它的进一步改进是非饱和 ReLU（Maas 等人，2013），其形式为：

$$a_{\mathrm{ReLU}}(t)=\begin{cases}t, & (t>0)\\ \delta t, & \text{其他}\end{cases} \tag{3.22}$$

其中 δ 是一个小的常数，这允许神经元呈现微弱的激活状态，无论神经元是否趋向激活都能产生小梯度。该方法的另一变体是非饱和 ReLU，它的 δ 与权重一起作为 BP 算法的学习参数。当 $\delta<0$ 时，会产生如图 3-12 的激活函数。此时，神经元会产生负信号，当神经元对预测无贡献时，也会使梯度移动更快。

另一个激活函数是 maxout（Goodfellow 等人，2013）。Maxout 是一个非常通用的激活函数，因此可以学习其形式。它能模拟从线性 ReLU 到二次方的任意激活函数，其定义如下：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

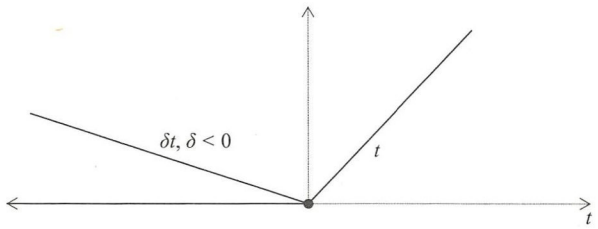


图 3-12　非饱和 ReLU

$$a_{\text{maxout}}(t_i) = \max_{j\in[1,k]} t_{i,j} \tag{3.23}$$

maxout 考虑 k 个相邻节点的输出，并将最大值作为激活函数值。它通过删除小特征减少特征数量。一定程度上，maxout 假定相近节点具有相似性，并选择活性最强的一个。maxout 是一个值得研究的激活函数，它的性能和价值还未被完全理解或实验证实。对 maxout 将会有更多的研究，欢迎读者深入这方面的讨论。

3.4.2　权重剪枝

网络层数太大或隐藏层的节点数量过多，MLNN 就会产生大量权重，或同等数量的自由度，这就可能生成一个过于复杂的网络系统。这会有两个缺点：计算复杂度高和容易过拟合。另外，在给定一个困难的学习任务，如何精确确定适合于该任务的低复杂度网络的最佳规模非常具有挑战性。通常的解决方法是从规模明显大于实际需要的网络开始，然后对网络进行修剪，逐步删除一些权重和节点来获得更简单的网络。

上述权重剪枝通常通过以下两种方法及其变体实现。第一种为启发式地去除对代价函数影响小的权重；第二种类型比较微妙，它在误差函数中

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

引入附加正则化，例如 L_1 项，这种方法更倾向于小的权重（实际上它会把一些权重置为 0）。目前，已有许多采用上述方法的技术（Castellano 1997；Lecun 1990；Reed 1993；Suzuki 2001）。此外，有人研究了如何处理权重剪枝网络的潜在的不规则性（Anwar 等人，2015）。

3.4.3 批量标准化

我们会在本书中研究学习深度网络的一些额外的技巧，包括一些正则化方法，如 dropouts；其他方法包括一些初始化技巧，如预训练和指导，其中最强大且常见的为批量标准化。

在图像输入网络之前进行标准化是常见的方法，通常将图像的像素值缩放为 $[-0.5, 0.5]$，均值为 0。在深度网络中，由于前一层的参数都会随着每次更新而变化，因此，每层的输入分布都会因不同的批量和样本而不同。这使深度网络的训练非常困难，特别是当激活函数趋向饱和时。

尽管我们假设同类所有样本是独立等同分布的，但这有时不正确。不同批次的样本，甚至同一类样本，它们的统计特性也不尽相同，我们称这种现象为协变量漂移（covariate shift），为解决该问题，我们标准化每一层的激活值。需要标准化的真正方差和需要减去的均值是未知的，但是可以从数据本身中估计出来。批量标准化就是其中一种方式，假设 z 是某一层的激活值，则：

$$z_{bn} = \frac{(z - \mu_z) * \alpha_z}{\sigma_z} \tag{3.24}$$

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

其中 μ_z 和 σ_z 分别为批量激活值的均值和方差。α 为网络待学习的参数之一，这可以看成是应用标准化后的学习延伸。α 也能在优化权重的过程中得到，即在多层网络上使用 BP 算法得到 α。批量标准化是一种强大的工具，即使采用非饱和激活函数，也能提高网络的学习速度。在图像可视计算中，批量标准化非常流行。

本章小结

本章通过回顾主要的发展历史，说明基本模型中的符号和术语来介绍前馈多层神经网络，以及 20 世纪 80 年代中期开始使用的人工神经网络“第二波浪潮”中的一些方法。另外，我们涵盖了近期的一些激活函数。最近的深度学习技术似乎是在神经网络的“第三波浪潮”中出现的，但早期发展中就已有其深厚的根基。

本章介绍的 MLP 很简短，但从讨论中可发现，神经网络方法主要关注两个问题：如何设计网络架构和如何设计学习算法。正如后面章节所述，最新文献给出了特定领域中针对这两个问题的解决方案。

参考文献

Anwar, Sajid, Hwang, Kyuyeon, and Sung, Wonyong. 2015. *Structured pruning of deep convolutional neural networks*. arXiv preprint arXiv:1512.08571.

Bryson, Arthur E. 1961. A gradient method for optimizing multi-stage allocation processes. *Proceedings of the Harvard University Symposium on Digital Computers and Their Applications*.

Castellano, Giovanna, Fanelli, Anna Maria, and Pelillo, Marcello. 1997. An

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

iterative pruning algorithm for feedforward neural networks. *IEEE Transactions on Neural Networks* 8(3): 519–531.

Cox, David R. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)* 20(2): 215–242.

Friedman, Jerome H and Stuetzle, Werner. 1981. Projection pursuit regression. *Journal of the American statistical Association* 76: 817–823.

Girosi, Federico and Poggio, Tomaso. 1989. Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation* 1: 465–469.

Goodfellow, Ian J, Warde-Farley, David, Mirza, Mehdi et al. 2013. Maxout networks. *International Conference on Machine Learning*, pp. 1319–1327.

Gunn, Steve R. 1998. *Support vector machines for classification and regression.* ISIS technical report.

Hahnloser, Richard HR, Sarpeshkar, Rahul, Mahowald, Misha A et al. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* 405: 947–951.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing et al. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034. IEEE.

Hosmer Jr, David W and Lemeshow, Stanley. 2004. *Applied Logistic Regression.* Hoboken, NJ: John Wiley & Sons.

Kelley, Henry J. 1960. Gradient theory of optimal flight paths. *ARS Journal* 30: 947–954.

Kolmogorov, Andrey N. 1956. On the representation of continuous functions of several variables as superpositions of functions of smaller number of variables. *Soviet Mathematics Doklady* 108: 179–182.

LeCun, Yann, Denker, John S and Solla, Sara A. 1990. Optimal brain damage. *Advances in Neural Information Processing Systems* 598–605.

Lingjaerde, Ole C and Liestol, Knut. 1998. Generalized projection pursuit regression. *SIAM Journal on Scientific Computing* (SIAM) 20: 844–857.

Lowe, David and Broomhead, D. 1988. Multivariable functional interpolation and adaptive networks. *Complex Systems* 321–355.

Maas, Andrew L, Hannun, Awni Y, and Ng, Andrew Y. 2013. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th International Conference on Machine Learning,* Atlanta, Georgia.

Minski, Marvin L and Papert, Seymour A. 1969. *Perceptrons: An Introduction to Computational Geometry.* Cambridge, MA: MIT Press.

Mohri, Mehryar and Rostamizadeh, Afshin. 2013. *Perceptron mistake bounds.* arXiv preprint arXiv:1305.0208.

Nair, Vinod, and Hinton, Geoffery. 2010. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning.* Haifa, Israel. 807–814.

Novikoff, Albert BJ. 1962. On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata*, pp. 615–622.

Powell, Michael James David. 1977. Restart procedures for the conjugate gradient method. *Mathematical Programming* 12: 241–254.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Reed, Russell. 1993. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks* 4(5): 740–747.

Rosenblatt, Frank. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65: 386.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. 1988. Learning representations by back-propagating errors. *Cognitive Modeling* 5: 386–394.

Suzuki, Kenji, Horiba, Isao, and Sugie, Noboru. 2001. A simple neural network pruning algorithm with application to filter synthesis. *Neural Processing Letters* 13(1): 43–53.

Vapnik, Vladimir, Golowich, Steven E, Smola, Alex et al. 1997. Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems*, 281–287.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

第4章
Convolutional Neural Networks in Visual Computing

卷积神经网络

在前几章中，我们介绍了全连接多层神经网络（MLNN），并用反向传播算法进行训练。一个典型的 MLNN 层由 n 个输入节点和 m 个神经元组成，学得 $n \times m$ 个参数或权重。尽管 MLNN 在某些问题上表现良好，特别是当不同维度的特征相互独立时，从中可能得到一些有用的特性。但是在一些情况下，例如输入数据的维度之间有强关联性或者 MLNN 的大小（层数和每层的神经元数）受到计算资源的限制，MLNN 就需要在网络架构上做出改变，以克服因数据或网络复杂性带来的这些限制。

尽管 MLNN 具有一般的表达能力，但我们希望能更好地利用特征间的局部依赖关系和不变性。特征维度的某些排序有特殊含义，相邻特征的某些变化也可能提供有价值的信息。一维（1D）特征空间，如语音识别中的数字音频信号；二维（2D）特征空间，如图像；三维（3D）特征空间，如视频，等等，我们都应当能在信号（图像、视频）的任意位置识别出这些模式。这些模式由一个模板表示，它本身可能是短信号。在前面的例子中，存在模板的数据在相邻样本上具有强关联性。此外，在第 1 章讨论的例子中，由卷积核得到图像的一些特征表示。这些都是将卷积层引入 MLNN 的动机之一（Fukushima，2003；Fukushima 等人，1983；LeCun 等人，1989）。本章研究引入卷积和池化层的 MLNN 及其组合来构造深度神

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

经网络。近几年，这些网络架构已经在一些长期存在的计算机视觉问题中取得了良好效果。

4.1　卷积与池化层

前面在线性回归中研究了稀疏性。稀疏性用权重矩阵或数组中的零的比例来定义。零权重或者神经元的响应值为零，分别表示从网络架构中删除传入或者传出的连接。

假定一个神经元的权重被强制稀疏化。所强加的稀疏性是随机的(称为 dropout)，并有一定结构。这种稀疏性通过以下方式使用：每个神经元的输入来自 k 个邻接的输入权重，邻接是重要的标准。在这种稀疏连通中，我们不使用全连接神经元的所有输出，仅让若干相邻的连接处于激活状态，完成连接的稀疏性和局部化。方便起见，考虑 $k=3$ 的情况。

如图 4-1 所示，由于每个神经元仅需输入 3 个相邻特征，所以它学习的是输入的局部特性。因此神经网络需要学习的特性可能是特征在局部空间改变的方式。这种局部性（locality）通过特定次序下信号维度的邻接性来定义。如果一组连接以信号顺序连接到邻接维度，则它是局部的。在图像中，对应于相邻的像素。因为图像像素按顺序排列，我们研究图像的局部性就是要发现这种顺序中像素之间的关系。除了图像，还有其他的应用，例如，音频信号按时间排序，而图像信号就是它本身像素的排序。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

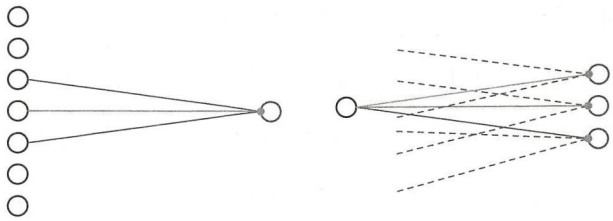


图 4-1 稀疏和局部连接

这种连接同时具有局部性和稀疏性。神经元学习到的特征具有局部性，其模板也仅作用于局部。因此，在音频或图像信号中，通过邻接特征的值的组合进行研究，如单词或声音；而在图像中，可以研究其局部特征，例如梯度、边缘或斑点。

如果每个神经元都有自己的局部连接，连接权重被看作用于特征检测的某个核，则局部连接意味着所有特征都是局部的。这在音频或图像中不一定正确。如果要在某个位置寻找一个模式，如边缘，我们需要在整个输入信号中寻找该模式。如果连接到图像某个局部位置的神经元学习检测到特定的边缘模式，我们希望学得的边缘检测器在图像其他位置也有效，能够找到任意有该模式的位置。换句话说，我们希望图 4-1 左侧所示的神经元不仅能连接到上述 k 个邻接输入特征，而且可以通过滑动或者移位，连接到其他 k 个其他相邻输入。我们无须为上述连接设置新的权重，而是使用相同的连接权重。这允许我们在信号的不同位置移动神经元并收集神经元的响应。一个神经元仅需 k 个权重就可以遍历整个信号。因此，我们仅需增加输出数量而不是权重数量。

如果排列适当，那么输出也是有序的。如果该神经元是边缘检测器，那么它会在图像的所有位置中寻找该模式。神经元的输出是与原始输入图

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

像大小大致相同的另一图像。输出是边缘存在或不存在的位置表示，这与第 1 章的边缘检测器类似。实际上，我们不会让神经元移动，而是很多神经元共享相同权重。事实上这两种思考连接建立的方式是等价的，我们仅考虑第二种解释——不同位置的神经元共享权值，这不仅保留了神经元的原始概念及其稀疏连接性，而且无须考虑移动神经元造成的额外复杂性。

假设输入为 d 维的特征向量 $\boldsymbol{x}=[x^{(1)},x^{(2)},\cdots,x^{(d)}]$，以及权重 $\boldsymbol{w}=[w_1、w_2、\cdots、w_k]$。从信号的开头开始，每次滑动一个特征，直到末尾，同时在每个位置上收集神经元的响应。该操作更正式的定义为：

$$z^{(j)}=\sum_{i=1}^{k}x^{(j+i-1)}w_k,\forall j=[1,2,\cdots,d-k+1] \tag{4.1}$$

上述操作称为卷积，更准确地说，这个过程称为互相关而不是卷积。卷积在严格的信号处理定义上，权重 w 需要以它的中心为轴翻转。为了方便起见，这两个操作都可用 $*$ 表示并可互换使用。但在阅读文献时，应注意这一点。上述定义是一维卷积，而不是第 1 章介绍的二维卷积。输出 $z(j)$ 是共享相同权重的神经元的输出响应。输出是神经元在信号不同位置的有序响应。由于许多神经元共享权重，因此输出的长度为 $n-k+1$。

这些神经响应会继续通过激活函数，如 tanh 或 ReLU。将这些激活的输出称为特征映射或激活。输出的特征映射的数量与共享权重神经元或滤波器的数量相同（Gardner，1988）。这些共享权重的神经元通常可互换地称为滤波器或核。术语滤波器是从传统的信号处理文献中借用的。图 4-2 描述了具有两个过滤器的架构。每个滤波器具有 $k=3$ 个权重。因为滤波

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

器在每次移动中从 k 个相邻输入位置收集信号，所以我们说滤波器的感受野是 k。多个这样的滤波器或一层的集合称为滤波器组（filter bank）。这些滤波器覆盖原始图像的区域通常被称为感受野。

图 4-2 中的神经元接收的是原始信号，感受野为 k。感受野的大小通常会引起争论。一种观点认为滤波器大小应当尽可能小，无论图像或信号的大小是多少均设置为 3 个像素。这对学习小规模特征及在网络中构建更深层的表示提供了可能。另一观点是在早期阶段学习稍大的滤波器，在较深的层次上学习较小的滤波器。这种争论尚未解决，需要根据应用需求设计适合的感受野。在本章后面的案例研究中，关于感受野大小的争论会越来越多。

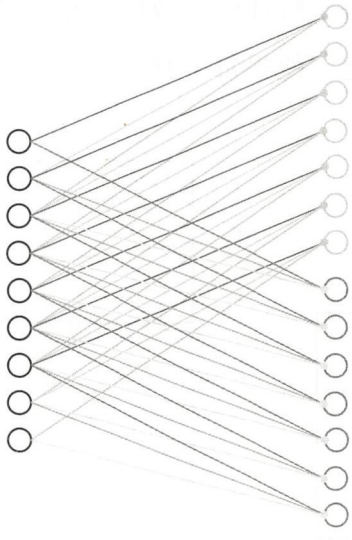


图 4-2 两组共享权重的稀疏神经元展示了步幅（stride）为 1 的卷积连接。滤波器大小为 3，特征映射大小（共享相同权重的神经元数量）为 7。线条的强度表示参数权重：颜色越暗，权重越低。类似地，不同阴影的圆圈表示神经元之间独特的权重分担。信号长度为 9，滤波器大小为 3，因此每个神经响应长度为 9－3＋1＝7

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

有两种方式来看待感受野。一种方式是仅把感受野看成前一层的接收信号。这种情况下，任何层感受野的大小可简单地设为 3。另一种方式是把感受野看作从输入层接收的信号，这意味着如果第一层滤波器组的大小为 3，第二层滤波器组的大小也为 3，那么第二组滤波器的感受野为 9。为简便起见，我们遵循第一种方式，但是要注意的是，更深的层对应着从原始接收信号的更大区域。

二维卷积层类似于一维卷积层，不同之处在于滤波器权重也是以二维排列的。图 4-3 的第一部分是典型的二维卷积层。每个神经元可以从二维信号的某个角（如左上角）开始，在一个方向上遍历，并在相反的角（如右下角）结束。每个输入本身可以是特征映射或图像通道的集合。在这种情况下，每个神经元与每个特征映射或输入通道进行卷积。每个通道的输出可以按位置相加在一起，使得每个神经元具有一个平均输出响应。这减

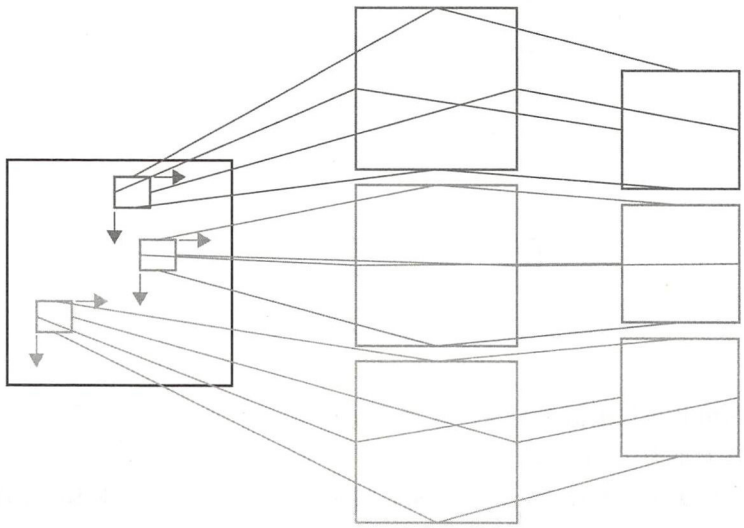


图 4-3　3 个神经元的卷积池化层

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

少了输出层的激活次数。假设第 a 层的输入具有 I 个通道（若来自其他图层，也可以是特征映射或激活），并且该层具有 L 个核 k，λ 表示激活函数。该层的输出经过激活函数的结果表示如下：

$$z^j = \lambda\left(\sum_{i=1}^{I} \boldsymbol{a}^{(i)} * \boldsymbol{k}_j\right) \forall j = [1,2,\cdots,L] \tag{4.2}$$

$*$ 表示在第 1 章中讨论的卷积运算。不同之处在于，这里的滤波器由神经网络学习得到，而不是在第 1 章中人工设计的。通过卷积层直接作用于输入图像上学到的滤波器类似于边缘或色斑检测器。这表示输入图像局部变化特征或类型。图 4-4 展示了从 CIFAR-10 数据集学习到的滤波器，及一个过滤器的激活函数输出（Krizhevsky 和 Hinton，2009）。

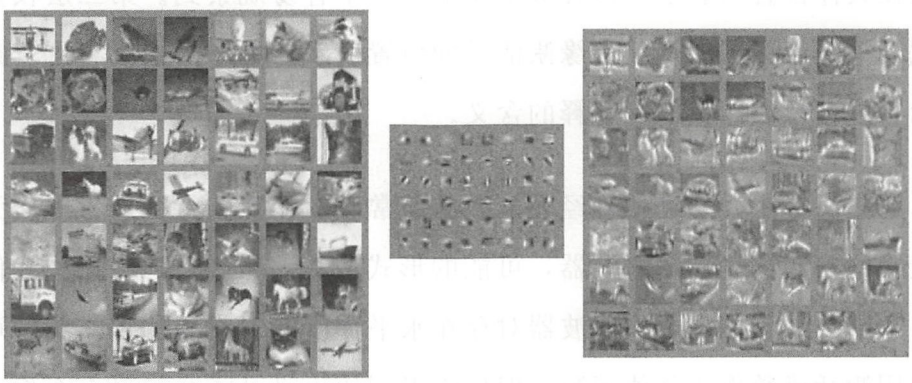


图 4-4 从左到右是来自 CIFAR-10 数据集的一些图像，从中学到的一些 5×5 二维卷积滤波器，以及左上滤波器产生的激活输出。用于学习此网络的代码可在 Yann 工具箱中找到（请参阅附录 A）

神经网络不只是学习概念的学习系统，还是分解数据并使用较小的信息片段表示数据的系统。使用这些表示，神经网络将数据映射到标签空间。例如，卷积神经网络（CNN）将图像分解为基本的小的图像组

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

件。这些图像组件通常是边缘和点，通常可在图像数据集中找到它们作为局部特性。通过在正确的位置以正确的方式使用这些边缘和点，并增加位移，就能够重新创建原始图像。我们将在 5.2.1 节研究这些细节。其他人利用该技术将图像映射到比标签空间更复杂的空间。考虑语义分割、对象定位或生成更高分辨率的图像，在这些应用场景下，除了网络预测（已经是图像）外，我们还应添加其他结构化模型，例如马尔可夫随机场，并产生结构化预测。在本章中，我们继续使用简单的标签空间。在更好的特征空间上连续变换和再变换之后，网络便能将图像映射到区分标签空间上。

在识别图像中存在什么类型的对象时，CNN 将图像分割成关于边缘或点在具体位置的表示。这种效果在第一层中容易观察到。第二层权重试图找到关于第一层的点和边缘激活后的位置表示。更深的层具有更复杂的表示，这通常没有我们可解释的含义。

在卷积层中，邻近的神经元被激活通常表示相似的特征。考虑作为特定边缘的模式检测器的滤波器，可能的形式为 [−1，0，1]。正如我们在第 1 章中已经看到的，该滤波器对存在水平梯度的点产生输出。我们可能会使用激活函数作为阈值函数，但最终得到的边缘可能会超过一个像素的厚度，它们并不能很好地符合边缘的定义。

由于卷积滤波器较小而所观察的范围较大，所以激活值的改变是平滑的。更陡峭的激活变换有助于更好的学习。为了实现这一点，通常卷积层之后有一个池化层或子采样层。这也能通过步幅卷积层（strided convolutioin）或步幅卷积和池化的组合来实现。池化通常被认为是实现这

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

一点的方法，尽管最近的研究使用步幅卷积来完全替代池化（Radford 等人，2015；Visin 等人，2015）。当我们移动模板的步幅超过一步时，卷积就变成了步幅卷积。

池化后，我们通过减小激活值的数量来降低了数据熵。缩减规模通常是有好处的，尽管会丢失一些激活值中的空间信息和空间频率，但是通过该层的滤波器组获得了更多的激活响应。更重要的是，池化允许我们在空间上利用及适应特征之间的某些不变性。当神经网络的深度越深，得到的激活值也就越多，激活值的规模也就越大。这将很快变得难以计算，而池化有助于我们保持可操作性。

从信号处理的角度来看，子采样的标准是平均池化，但是在 CNN 中通常优选最大池化。实现大小为 p 的最大池化，选择 $p \times p$ 的滑动窗口。类似于卷积，这个滑动窗口可以是大步幅或步幅为 1。通常，池化的步幅与池化窗口大小相同（$p \times p$）。在窗口中选择最大值，并用该值表示整个窗口。最大池化值代表最强的响应，在图像处理问题中，最强的响应对应于滤波器要查找的最佳匹配模板。第 5 章将尝试解决并讨论池化会丢失信息的问题。

图 4-3 显示了典型的栈式卷积池化或卷积池化层（convpool layer）。在卷积池化层中，权重变小，但有大量的表示其特征的特征映射。通常，滤波器较小但大于步幅。池化窗口大小通常小于或等于卷积滤波器大小。尽管它们在文献中可以称为不同的层，但卷积池化层常作为同一层来实现。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

4.2　卷积神经网络

卷积神经网络（CNN）是一种将信号输入到栈式卷积池化层的神经网络，最后一层为全连接层，并通过 softmax 层得到输出（见图 4-5）。

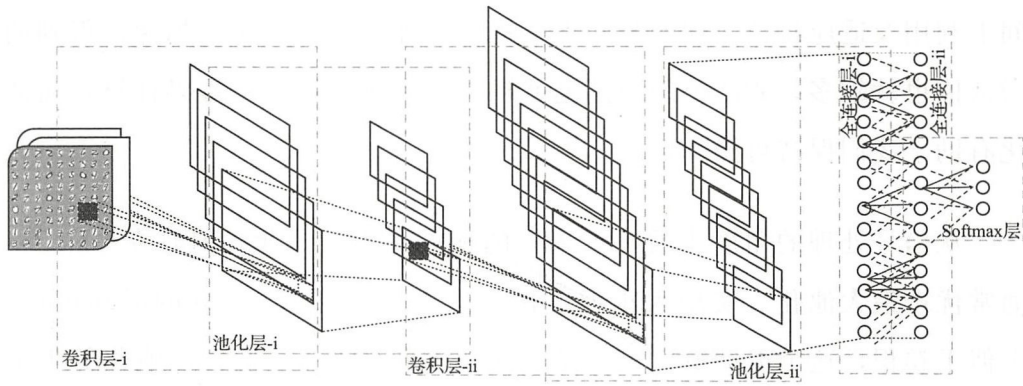


图 4-5　完整的卷积神经网络（LeNet）

类似的初始设置可应用于自编码器和其他神经网络，但现在我们只考虑用于分类的网络，在第 5 章中重新讨论其他的网络架构。在 softmax 层前的全连接神经网络可以有卷积层，也可以没有。由于卷积池化层是可微的，因此 CNN 中的学习与 MLNN 中的学习是相同的，这在第 3 章中已经介绍。我们研究的其他优化过程也适用于 CNN。

上述构建 CNN 的基本思想显而易见，但实践中，可以选择不同的方式设置网络，从而根据特定问题形成不同的网络架构。接下来，我们研究几个知名的网络架构，希望说明一些构建特定 CNN 的典型方法。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

案例学习

既然已经创建了CNN，就研究一下热门CNN的架构及基于CNN的应用。

案例0：MNIST数据集

进入第一个网络架构之前，学习一个最常用的计算机视觉数据集：手写体字符MNIST数据集。了解该数据集有助于我们研究第一个CNN架构，因为它就是为这个数据集的应用而设计的。

MNIST是一个灰度图像数据集，取代了黑白图像NIST数据集。MNIST包含紧密剪裁的手写体数字字符图像。每幅图像包含28×28个像素，中间是一个字符。数据集有70 000幅图像，大致均匀地分布在所有类中。总共有50 000幅图像用于训练，10 000幅用于测试，10 000幅用于验证。这些划分预先完成且一直保持。图4-6显示了此数据集的一些示例图像。

图像大小、比例和书写的一致性使得该数据集成为在初始阶段研究和测试机器学习算法的优秀工具。由于数据集安排得很好，在使用CNN时我们可以观察到每个过滤器的行为，使它成为研究CNN的理想数据集。遵循CNN的传统，我们从MNIST数据集开始研究CNN。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

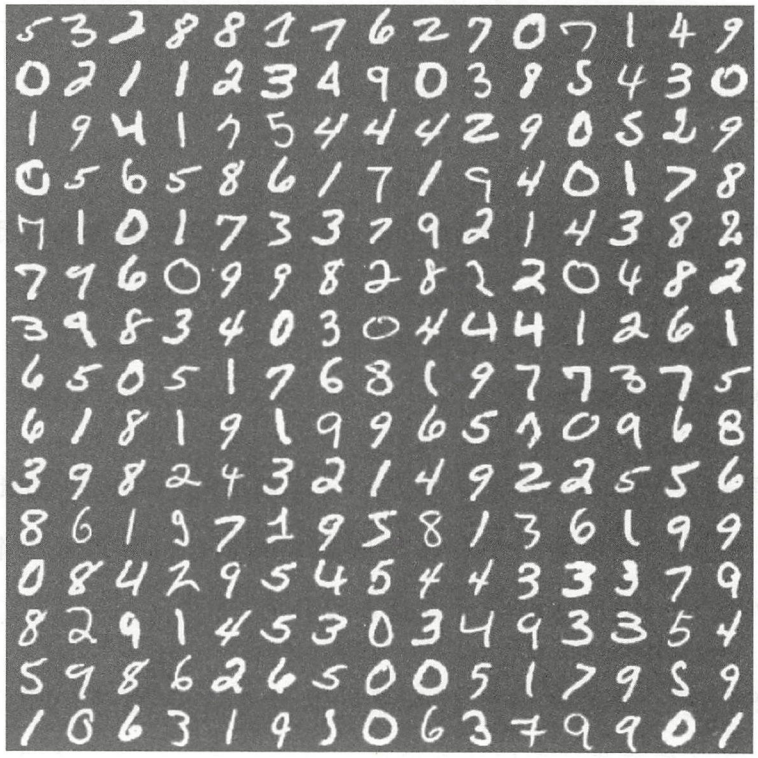


图 4-6　MNIST 数据集的示例图像

案例 1：LENET

Yann LeCun 教授提出的数字识别 CNN 是最早也是最受欢迎的 CNN 之一（LeCun 等人，1998 年）。该网络称为 LeNet5 或简称为 LeNet。LeNet5 是有两个卷积层和一个全连接层的 CNN，详细架构设置见表 4-1。

用于 MNIST 的 LeNet5 是一个 28×28 单通道灰度图像，并作为第一层的输入。第一层使用 5×5 的感受野。最初，由于计算的原因，它只有 6 个特征映射，但是随着当前技术的发展，现在这个网络的第一卷积层

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

已有 20 个特征映射。因此该层输出的 20 个特征映射的大小为 28－5＋1＝24（×24）。对其做步幅为 2 的池化，则下一个卷积池化层的输入为 12×12。下一层学到 50 个特征映射，每个特征映射大小为 8×8，通过下采样到 4×4。在到达全连接层前，将激活函数的输出调整为 1×800，然后输入到由 500 个神经元组成的隐藏层。最后通过 10 个节点组成的 softmax 层输出，每个输出节点对应一个数字字符。在 LeNet5 中，所有激活都是能提供更快学习的 ReLU 单元。图 4-7 显示了 LeNet5 卷积层的滤波器及其激活输出，这是以 0.01 的学习率对网络进行 75 次迭代训练的结果。经过 CNN 的训练，在 MNIST 数据集上产生 99.38 %的准确度。生成这些结果和数字的代码可在 yann 工具箱的教程中找到，详细信息请参阅附录 A。

表 4-1 现代版 LeNet 的 MNIST 分类

层号	输出大小	感受野	特征映射数	神经类别
1	28×28×1	5×5	20	卷积
2	24×24×20	2×2	—	池化
3	12×12×20	5×5	50	卷积
4	8×8×50	2×2	—	池化
5	800	—	500	全连接
6	500	—	10	Softmax

我们对 MNIST 数据集和 CNN 原型 LeNet5 有了一定的了解，下面继续介绍几个现代 CNN。这些 CNN 对计算机视觉问题技术的复兴产生了重要影响。这种复兴是由多种因素带来的，包括计算技术的进步（强大的 CPU，但更重要的是支持并行计算的 GPU）、更大和更好的图像数据集、开放视觉对象分类（VOC）、ImageNet 竞赛等（Russakovsky 等人，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

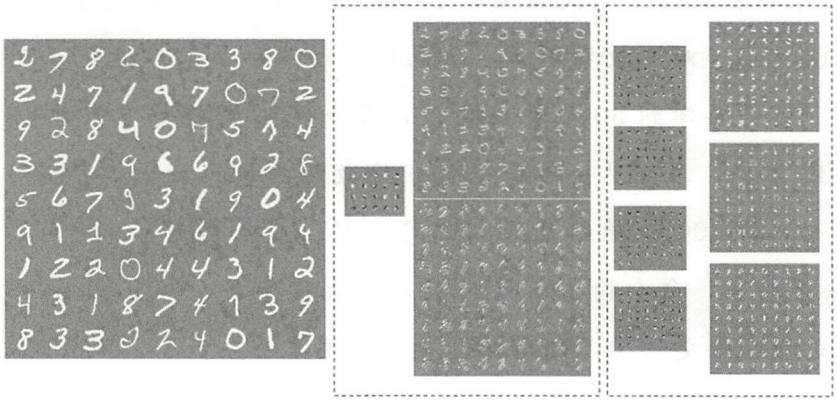


图 4-7　MNIST 图像 LeNet 卷积池化层中的滤波器和激活

2015)。由于这里的每个网络都代表了当时的重大进展，因此能很好地说明改进 CNN 架构的基本思想。

案例 2：ALEXNET

Krizhevsky 等人在 2012 年神经信息处理系统会议（NIPS）上介绍了他们的 CNN（Krizhevsky，2012），称为 AlexNet。该网络由 2012 年 ImageNet 挑战赛的获胜者（Russakovsky 等人，2015）发明。

ImageNet 与 MNIST 一样，是神经网络的第三波浪潮中众多研究者们依赖的大规模数据集。在 AlexNet 时代，ImageNet 是一个拥有超过 22000 个类别、1500 万个图像的数据集。ImageNet 竞赛，即 ImageNet 大规模视觉识别挑战（ILSVRC）使用了其中大约 1000 个类别。ImageNet 类别较为复杂，甚至连人类也很难进行完美分类。在此数据集上，正常人的平均分类准确率约为 96%。使用常规技术的识别系统，如第 1 章中讨论的那些，最多只能达到 75 %的准确率，2012 年 AlexNet 是第一个打破 80 %准

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

确率的系统（Russakovsky 等人，2015）。

AlexNet 的核心只是简化了深层的 LeNet。图 4-8 中的顶层网络显示了 AlexNet 的整个体系架构。它有五个卷积池化层，后面是两个全连接层和 1000 个节点的 softmax 层，一个节点对应一个类别。相比之下，这个网络的体系架构像 LeNet 的扩展和深化，但参数的数量和模型复杂性导致了大量的过拟合。即使 ImageNet 数据集有 1500 多万个图像，也无法以稳定的方式训练该网络。计算上，训练如此规模的网络也是一个主要问题。在实现时，AlexNet 是最大和最深的神经网络之一。除了使用随机梯度下降的 CNN 基本训练算法外，必须采用与多种新技术相结合的方式，使该网络在训练时不至于过拟合。下面详细讲解这些技术。

AlexNet 的一个创新是选择 ReLU 激活，而不是传统的 tanh 或 sigmodic 激活（Nair 和 huntin，2010）。tanh 和 sigmoid 激活是饱和激活，而 ReLU 不像其他激活函数那样在 1 附近饱和，因此在接近 1 时仍然具有较大梯度，这有利于提高学习速率。

AlexNet 的另一个重要创新是使用多 GPU 学习。最初的实现仍然可以作为英伟达卷积神经网络（cuda-convnet）使用，但从那以后，高级版本已经以许多工具箱的形式实现。图 4-8 中的下半部分图像把网络被分成两部分，这是因为网络的一部分在一个 GPU 上运行，另外一部分在另一个 GPU 上运行。由于 Krizhevsky 等人最初在两个内存为 3GB 的 Nvidia GTX 580 GPU 上训练网络，一个 GPU 无法容纳整个网络。这种拆分可使网络学习到更专一的特征。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

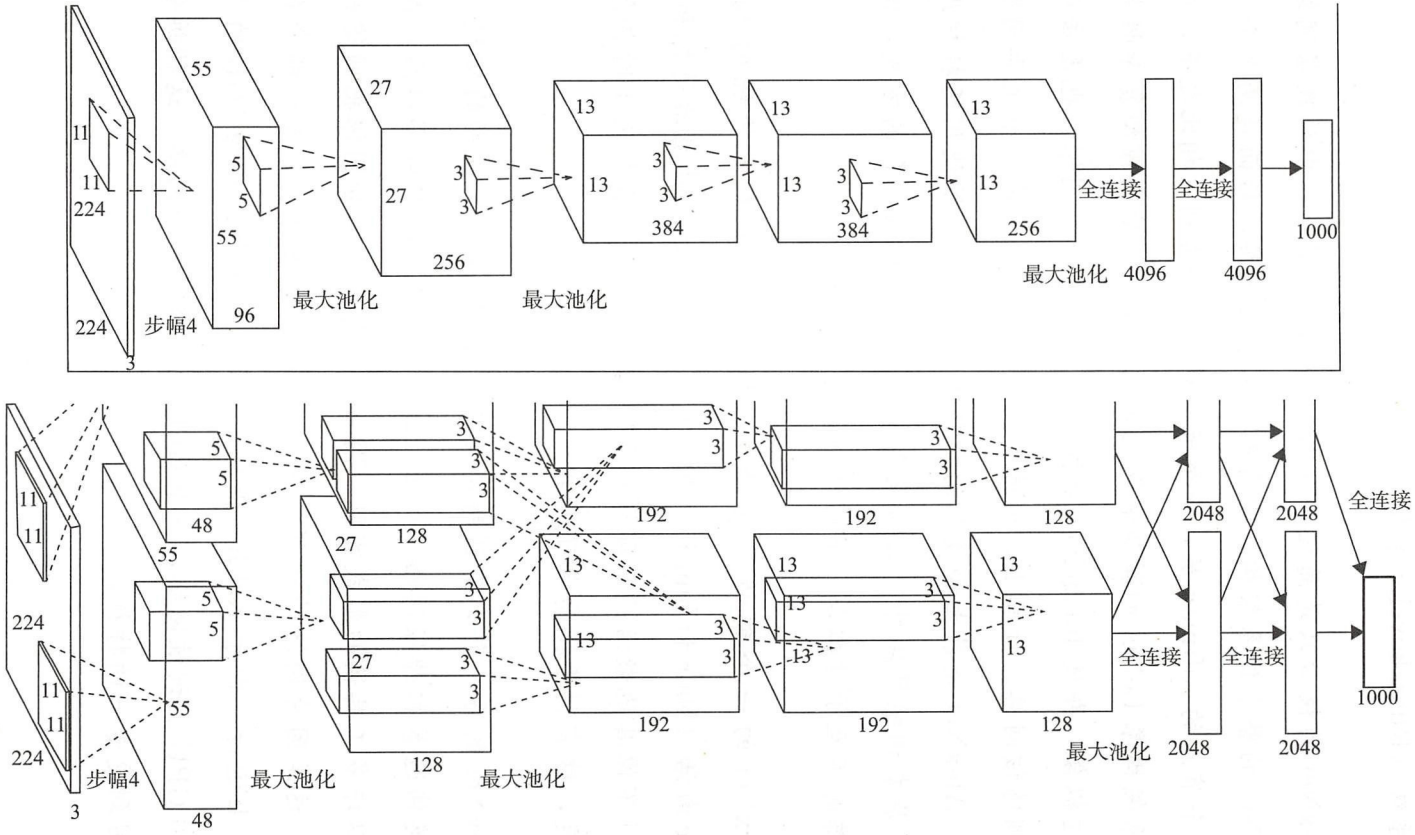


图4-8 ALEXNET结构（Alex Kirzhevsky提供）

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

图 4-9（上图）展示了训练过程中学到的滤波器。前三行通常在一个 GPU 上学习，学得的特征对颜色不敏感，而后三行是在另一个 GPU 上学到对颜色敏感的特征，这种行为要通过多次重复才能得以表现。虽然网络在两个并行的 GPU 中训练，但它们不是完全独立的多列网络，而是会相互通信。

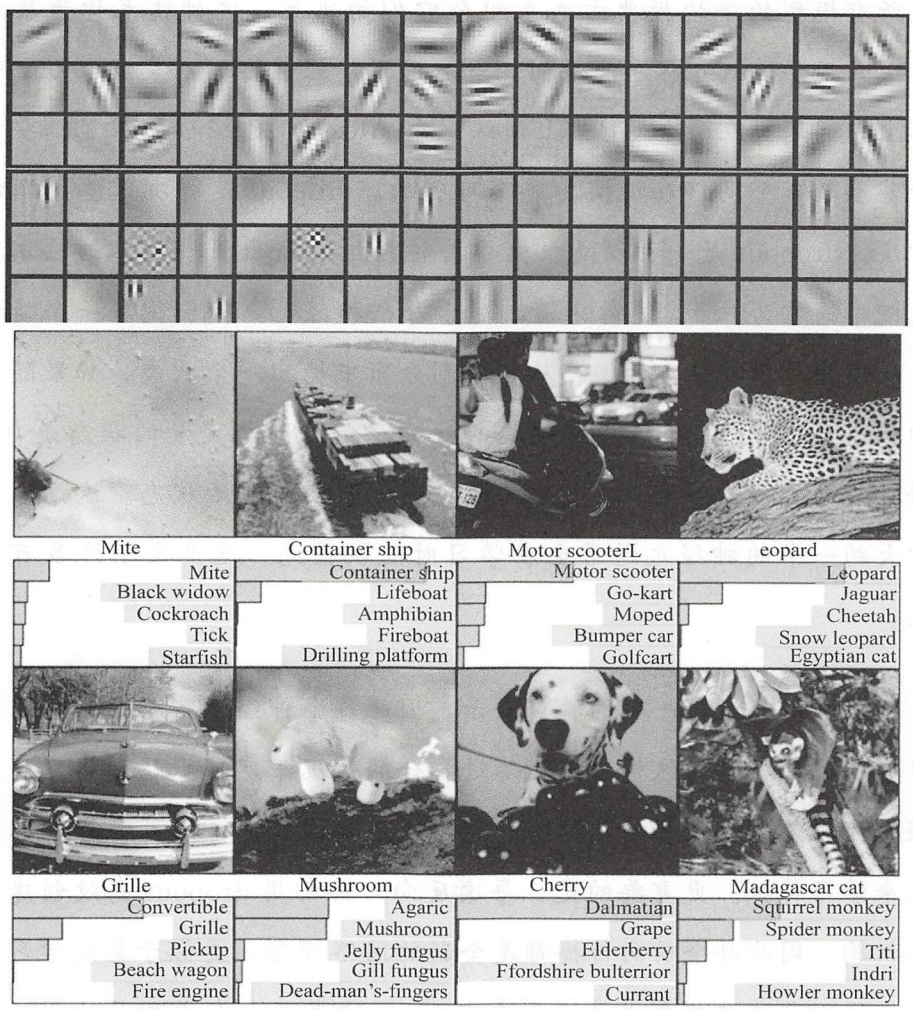


图 4-9　AlexNet 第一层学习到的滤波器、采样及预测结果（由 Alex Kirzhevsky 提供）。这个图像是灰度级的。这个图像的彩色版本可以在原始文章中找到（Krizhevsky 等人，2012）

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

AlexNet 的另一个重要技术贡献是 dropout。AlexNet 是一个具有大量参数的神经网络，如果不进行合适的正则化，训练这样大规模的网络很容易造成严重的过拟合。dropout 是一种避免特征映射协同适应（coadaption）的方法（Srivastava 等人，2014）。为提高性能，我们会训练多个模型，并通过所有模型的平均投票方式来组合它们的决策。这种技术称为集成学习，通常用于随机森林和决策树。当应用到神经网络时，集成学习会带来计算上的问题。在两个 GPU 上训练一个 AlexNet 已经很困难了；要训练数十甚至数百个网络在现实中是不可行的。dropout 是解决这些问题的一种方法。dropout 是一种以随机概率将神经元的输出强制为零的技术。该随机概率通常为 0.5。这意味着，对于每个神经元，我们关联一个随机变量，从该变量中以 0.5 的概率从伯努利分布中输出一个状态。如果随机变量的状态是 1，我们允许信号通过神经元；如果随机变量的状态是 0，我们将神经元的输出降到 0，而不管神经元是否处于激活状态。这意味着网络中大约一半的神经元通常会在学习时随机关闭。这产生了许多有利的效果。

由于网络只运行一半的特征表示，因此每个神经元需承受两倍压力来学习有意义的特征。这样可以在不过拟合的情况下实现更快的学习。由于神经元随机地关闭或开启，这些神经元不再协同适应，而是独立于其他神经元来学习特征。更重要的是，每次反向传播都用 dropout 高效创建新的体系机构，因此每个前向传播都是全新的网络模型，类似于集成学习，其中一个区别是这些模型共享参数。dropout 通常不会在卷积层使用，因为它会删除整个特征映射。dropout 应用于单个特征响应，这些响应随机 dropout 特征映射中的少数位置，这本身并不理想。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

表 4-2 详细显示完整的 AlexNet 体系架构。AlexNet 所产生的结果在当时是极具影响的，然而这种体系架构受到了计算和存储容量的限制。作者预测更深更复杂的模型将带来更高的精度。这个预言已经被多次证实，并产生了一些其他更新的网络架构，我们将在本章其余部分和第 5 章中讨论。

除了赢得比赛和显著提高 ImageNet 数据集准确性外，AlexNet 还展示了一些有趣的主观结果，如图 4-9 所示。不仅标签的第一选择是最佳的，而且系统预测的其他标签在语义上也有很大关联（在某些情况下，这纠正了数据集本身的标签错误）。举例来说，雪豹和美洲豹是豹的额外猜测，它们不是很糟糕的备选方案。这样的主观结果似乎表明，AlexNet 可能已经学会了以更语义的方式来表示底层对象，从而将语义相关的对象分组在一起。

表 4-2 AlexNet 体系架构

层号	输出大小	感受野	特征映射数	神经类别
1	224×224×3	11×11，步幅 4	96	卷积
2	—	3×3，步幅 2	—	池化
3	55×55×96	5×5	256	卷积
4	—	3×3，步幅 2	—	池化
5	13×13×256	3×3，padded	384	卷积
6	13×13×384	3×3，padded	384	卷积
7	13×13×384	3×3	256	卷积
8	30 976	—	4096	全连接
9	4096	—	4096	全连接
10	4096	—	1000	Softmax

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

案例 3：GoogLeNet 和 inception 模块

在 2015 年，谷歌提出了特殊的卷积层，称为 inception 层（Szegedy 等人，2015）。AlexNet 类型的网络有一个基本缺点：在提供深度的同时计算成本很高。inception 模块可使网络具有更少的参数，同时提供更深的深度。inception 模块是一个网中网系统。图 4-10 显示了一个 inception 模块。每个 inception 模块包含几个并行的卷积层，这些卷积层通过 1×1 卷积层实现降维。有一个最大池化层。这些层在传递到另一个 inception 模块或正则化模块之前被串联起来。1×1 卷积降低了维数，类似于一种称为嵌入的技术。这被传递到更昂贵的 5×5 层和 3×3 层。嵌入层在传递到下一层之前执行簇操作。胶囊是一种相似或相关的思想，它向前一步，进行了一个霍夫变换，类似于聚类投票（Hough transformation 等人，2011）。

GoogLeNet 是一个使用 inception 模块的特定网络。不加上池化的网络层数为 22 层。这样的深度网络，会出现梯度消失问题。梯度消失问题是由深度带来的，因为误差不足以产生足够强的梯度，无法在任何方向上移动权重。为了避免这种情况，在它们之间，层被分支成一个全连接的 softmax 层。这些分类器层不是为了在分类精度上执行得更好，而是增加更多的误差，使得一些梯度产生额外的区分特征。在网络之间增加额外的 softmax 层，这些节点产生的错误有助于更强的有监督性，从而产生更强的梯度。实际上，我们在正则化网络，使其所有层均具有差异性。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

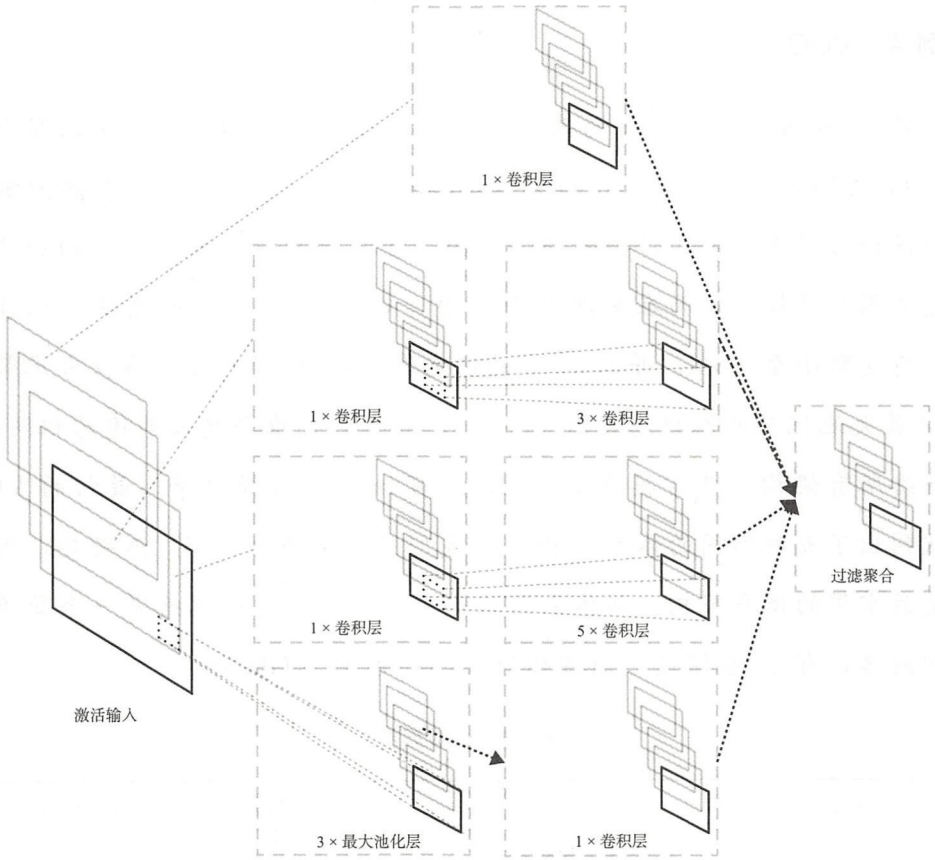


图 4-10 GoogLeNet 的 inception 模块

GoogLeNet 的一个特点是一开始有传统的卷积层，然后是 inception 模块。传统的卷积层产生的激活使用 1×1 层时容易聚集。与 AlexNet 不同（0.5 概率的 dropout 和 0.9 动量），GoogLeNet 的全连接层用 0.7 概率的 dropout 训练。这里不展示 GoogLeNet 的效果，留给感兴趣的读者查阅原始参考资料。GoogLeNet 能够显著提高包括 ImageNet 在内的几个数据集的技术水平。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

案例 4：VGG-19

VGG 网络是一个流行的基准网络，在 2014 年的 ImageNet 比赛中击败了 GoogLeNet（Simonyan 和 Zisserman，2014）。VGG 网络在滤波器大小的设计哲学上与前面讨论的网络不同。在之前的所有网络中，初始几层的滤波器尺寸较大，而在更深层则较小。VGG 遵循相同的想法，但开始时的感受野小至 3×3，并以 3×3 的恒定速度增长。VGG 网络仅考虑带有 2×2 最大池化层的卷积滤波器（3×3），并将它们堆叠起来创建卷积网络。该网络所有架构中非常接近 LeNet，没有 inception 模块等。最初的 VGG 文献列出了五种不同的体系架构，但我们只对第五种（E）感兴趣，因为它是最常用的体系架构。该体系架构如表 4-3 所示。E 网络是一个简单的卷积网络，有 24 层深度，通常称为 VGGNet 或 VGG-19。

表 4-3　VGG 网络

层号	感受野	特征映射数	神经类别
1	3×3，步幅 1	64	卷积
2	3×3，步幅 1	64	卷积
3	2×2，步幅 1	—	池化
4	3×3，步幅 1	128	卷积
5	3×3，步幅 1	128	卷积
6	2×2，步幅 1	—	池化
7	3×3，步幅 1	256	卷积
8	3×3，步幅 1	256	卷积
9	3×3，步幅 1	256	卷积
10	3×3，步幅 1	256	卷积
11	2×2，步幅 1	—	池化

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

（续）

层号	感受野	特征映射数	神经类别
12	3×3，步幅 1	512	卷积
13	3×3，步幅 1	512	卷积
14	3×3，步幅 1	512	卷积
15	3×3，步幅 1	512	卷积
16	2×2，步幅 1	—	池化
17	3×3，步幅 1	512	卷积
18	3×3，步幅 1	512	卷积
19	3×3，步幅 1	512	卷积
20	3×3，步幅 1	512	卷积
21	2×2，步幅 1	—	池化
22	—	4096	全连接
23	—	4096	全连接
24	—	1000	Softmax

使用相同大小的较小滤波器来构建大型网络层的策略最初是为了方便分析深度对性能的影响。正如预期的那样，网络越深性能越好。尽管 VGG 的研究结果声称适用于各种大小的层，但最近的一些结果挑战了“更深更好”的说法，因此目前还不能给出一个明确的结论。作为第一层上的 3×3 滤波器，可以很好地观察网络学习到的各种检测模板。图 4-11 显示了 VGG19 网络第一层的所有滤波器。请注意，滤波器主要是角点检测器、边缘检测器和点检测器，它们是构建更高层特征的基本构造块。VGG19 已经成为一个原型网络，它的第一层特征进一步证明了 CNN 可以用来学习特征检测器。

最初工作中，为了训练 VGG 网络，使用了 0.9 动量。当性能饱和时，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

就改变学习速率。训练分 74 轮（epochs）完成，每批（batch）训练样本大小为 256。利用这个网络架构，VGG 成功地击败了 GoogLeNet。

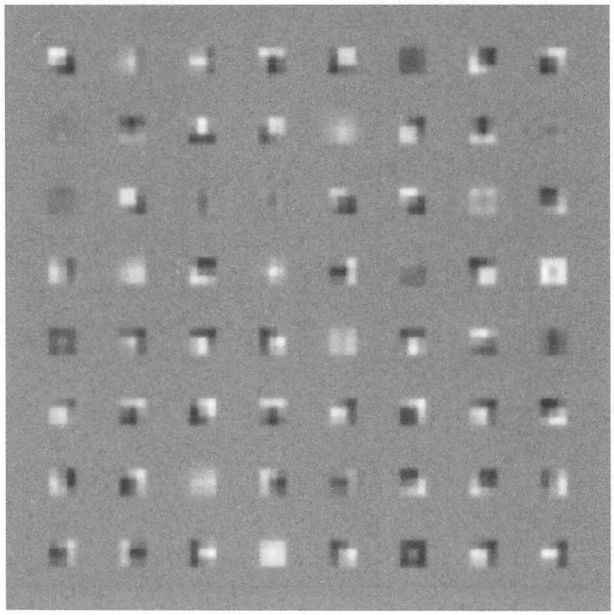


图 4-11　VGG-19 网络的第一层特征。参考网站 convolution. network 查看彩色图

需注意到，VGG 的层比 GoogLeNet 浅，这导致许多人认为，较小且大小一致的滤波器即使不是更好的技术，也是一种好的技术，这仍然是一个开放的争论问题。

案例 5：残差网

在 GoogLeNet 中，发现了一个层的体系架构，该层分成几个分支，并生成求和或级联输出。以求和或级联结束的层将梯度拆分为相等的部分，并在反向传播期间传播梯度。受 GoogLeNet 的启发，He 等人提出 ResNet。ResNet 是一种新颖的体系架构，它将一个层分成两个分支，其

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

中一个分支对信号不做任何处理，而另一个分支像典型的层那样对信号进行处理（He 等人，2015）。未处理的数据或残差被添加到未经改变网络的原始信号中。这将会使网络分裂，其中一个分支非常简单地传播梯度，而不改变梯度。这让更深层的网络学习使用很强的梯度。在 GoogLeNet 和 ResNet 中，数据（以及梯度）都直接通过，但 inception 模块仍有池化层，ResNet 模块则完全去掉了池化层。图 4-12 验证了这个想法。注意，有一些文献证明，非常深的 ResNet 相当于浅的递归神经网络（其中一个层以其过去的信号作为反馈输入，可以有也可以没有存储记忆状态）（Liao 和 Poggio，2016），并且提出了类似结构，称为高速公路网，其灵感来自递归神经网络（Srivastava 等人，2015）。

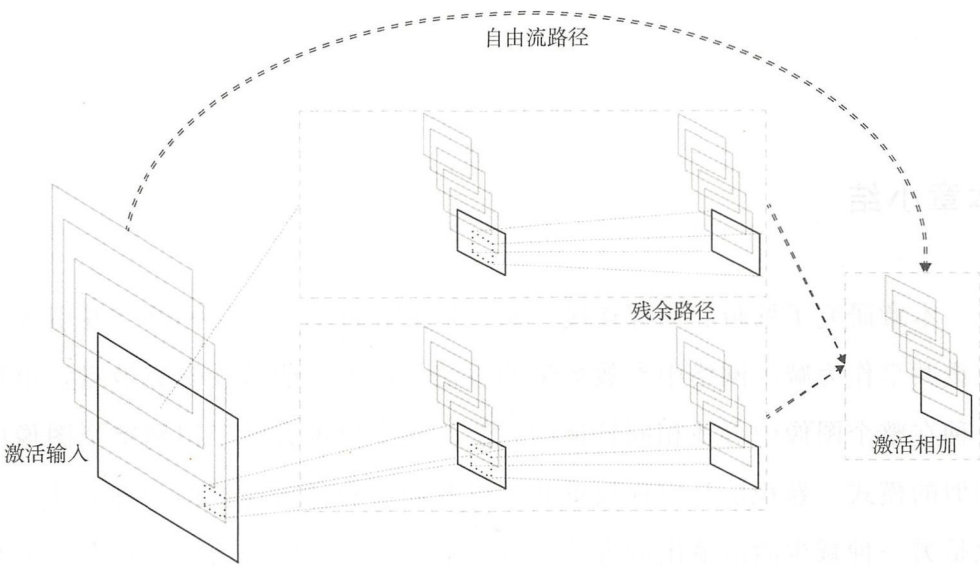


图 4-12 典型的残余层结构。有几个（一个）模块是参数化的，并附有不影响梯度的连接。图层的最终输出是使用图层的原始输入生成的所有激活的总和

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

对原始ResNet体系结构提出了若干改进，一种是创建体系架构，如一些层可以在训练时间期间随机关闭，并且在测试期间得到补偿，这非常类似于dropout（Huang等人，2016）。此网络架构将使用标识替换dropout模块（这意味着剩余部分的处理部分仅产生零）。虽然在训练时丢弃层是违反直觉的，但是必须注意，在测试时网络仍然是深度网络，并且在训练时具有随机深度（深度是随机的）。任意深度的网络往往会导致过拟合，有时比具有较少层的网络产生更差的性能，但ResNet却随着网络深度增加而提升性能。ResNet击败了前面讨论过的所有网络设置的性能记录，其中的一个变体在撰写本书时保持了ImageNet挑战赛的记录。赢得ImageNet竞争的ResNet最多有152层，也在CIFAR－10数据集上实现了1000层的网络。

本章小结

本章研究了稀疏和局部连接，而不是第3章的全连接网络。并引入了权重共享作为减少网络中参数数量的方法。这不仅能减少参数数量，也是一种在整个图像中寻找相同特征的方法。共享权重有助于识别整个图像中相似的模式。卷积层是实现权重共享以及局部和稀疏连接的一种方法。池化是另一种减少激活输出的方法。由于卷积层产生的激活输出在存储上是低效的，因此池化很重要，这还帮助我们使用了空间不变的特性。

利用卷积和池化层来创建卷积池化层。卷积池化层是现代CNN中枢神经系统，用于神经计算机视觉的主要工具。我们同时研究了卷积池化层

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

的各种性质。注意到CNN卷积池化层的最初几层学习类似于边缘检测器或点检测器的滤波器。我们还介绍了几种流行的CNN，如LeNet、AlexNet、GoogLeNet（InceptionNet）和VGG-19。在这个过程中，研究了这些网络使用的一些特殊层或技术，包括dropout、滤波器大小、inception模块，及ResNet体系结构，特别是为更深层网络的稳定学习提供可自由流动的梯度。

有趣的是，有研究表明灵长类视觉皮层中可能存在类似于CNN（Hubel和Wiesel，1968）中卷积处理的过程，特别是在感受野的概念上。尽管我们提醒读者注意这种解释，但CNN中有许多技巧（如dropout训练），可能仅有计算意义，而不一定有任何对应的生物意义。

参考文献

Fukushima, Kunihiko. 2003. Neocognitron for handwritten digit recognition. *Neurocomputing* 51: 161–180.

Fukushima, Kunihiko, Miyake, Sei, and Ito, Takayuki. 1983. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics* 13(5): 826–834.

Gardner, Elizabeth. 1988. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General* 21: 257.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing et al. 2015. *Deep residual learning for image recognition*. arXiv preprint arXiv:1512.03385.

Hinton, Geoffrey E, Krizhevsky, Alex, and Wang, Sida D. 2011. Transforming auto-encoders. *International Conference on Artificial Neural Networks*, Springer, pp. 44–51, Espoo, Finland.

Huang, Gao, Sun, Yu, Liu, Zhuang et al. 2016. *Deep networks with stochastic depth*. arXiv preprint arXiv:1603.09382.

Hubel, David H and Wiesel, Torsten N. 1968. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology* 195: 215–243.

Krizhevsky, Alex and Hinton, Geoffrey. 2009. *Learning multiple layers of features from tiny images*: Citeseer.

Krizhevsky, Alex, Sutskever, Ilya, and Geoffrey, Hinton E. 2012. ImageNet

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*: 1097–1105.

LeCun, Yann, Boser, Bernhard, Denker, John S et al. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1: 541–551.

LeCun, Yann, Bottou, Leon, Bengio, Yoshua et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.

Liao, Qianli and Poggio, Tomaso. 2016. *Bridging the gaps between residual learning, recurrent neural networks and visual cortex*. arXiv preprint arXiv:1604.03640.

Nair, Vinod and Hinton, Geoffery. 2010. Rectified linear units improve restricted Boltzmann machines. *International Conference on Machine Learning*, Haifa, Israel, pp. 807–814.

Radford, Alec, Metz, Luke, and Chintala, Soumith. 2015. *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv preprint arXiv:1511.06434.

Russakovsky, Olga, Deng, Jia, Su, Hao et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115: 211–252.

Simonyan, Karen and Zisserman, Andrew. 2014. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

Srivastava, Nitish, Hinton, Geoffrey E, Krizhevsky, Alex et al. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15: 1929–1958.

Srivastava, Rupesh K, Greff, Klaus, and Schmidhuber, Jürgen. 2015. Training very deep networks. *Advances in Neural Information Processing Systems, 2377-2385*.

Szegedy, Christian, Liu, Wei, Jia, Yangqing et al. 2015. Going deeper with convolutions. *Proceedings of the IEEE, Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA.

Visin, Francesco, Kastner, Kyle, Cho, Kyunghyun et al. 2015. *ReNet: A recurrent neural network based alternative to convolutional networks*. arXiv preprint arXiv:1505.00393.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

第5章
Convolutional Neural Networks in Visual Computing

卷积神经网络的新进展

在第4章中，我们讨论了几种用于图像分类的卷积神经网络（CNN）。由于这些网络的实现较为直接，因此称为一般的CNN。因为卷积池化层和神经架构具有通用性，因此它们的应用范围很广，不局限于简单的图像分类问题。目前，我们仅研究了监督学习。CNN非常适合解决其他问题，因此一直是活跃的研究领域。CNN包括很多它的变体，以及网络架构的附加组件，正以各种新颖的方式被广泛应用，包括目标定位、场景检测等。尽管上述应用都很重要，但本章仅研究卷积池化层和CNN架构在计算机视觉领域的应用。

上一章中构建了几个分类网络，我们发现，CNN中较浅层提取的图像表示是构建检测器的一些基本组件，如边缘和点。我们同时也发现多个数据集在这些层上提取的特征类型是相似的。这种表示上的相似性引发了一个问题，从一个数据集中学习到的网络是否可用于对另一个数据集的分类操作？简单的回答是“可以的”。我们在第4章中看到的一些网络都有对应的权重和网络架构，如ResNet、VGG-19、GoogLeNet和AlexNet，这些网络是经过预训练的、可下载的、可直接使用的。训练大型网络的任务通常复杂困难，比如，由于权重爆炸、梯度消失、不合理的正则或者内存、计算能力不足导致的网络崩溃，以及学习速率等超参

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

数的选择等。深度学习社区在发布预训练网络和允许别人使用这些网络上很有帮助，任何期望在小数据集上训练深度网络的人都能用其作初始化，并根据任务进行微调。微调预处理网络已成为神经计算机视觉领域的常见做法。大多数情况下，这些网络是使用 ImageNet 数据集进行预处理的。那么，在一个不同的数据集，甚至在统计意义上与 ImageNet 相悖的数据集中使用来自这些现成网络的表示进行初始化然后进行微调是否有效？接下来展开讨论。

5.1　预训练网络

图像数据集的复杂度不断提高，前 10 年类似 MNIST 这样的数据集已被结构化和控制到不自然的程度。最近的数据集像 ImageNet、PASCAL 或 Caltech-101/256 等包含的是由个人相机拍摄的图像（Everingham，n. d.；Fei-Fei 等人，2006；Russakovsky 等人，2015）。这些视觉对象分类（VOC）数据集，即自然图像数据集——不设置严格控制，而是现实世界直接拍摄的照片。由于这些数据集都包含自然图像，因此可能具有相似的统计特性，因而使用它们的 CNN 可能产生相似的特征表示。

由于有些数据集比其他的拥有更多图像或类别，因此，利用这些数据集有更大的潜力学到更多的图像特征。由于这些数据集中的图像都很相似，因此一个数据集上训练的网络可作为另一个数据集的特征提取器，不再需要后期显式的学习或调优。假设我们有两个数据集 A 和 B，A 的图像数量和类别比 B 大得多。如果仅通过用 B 构建 CNN 来学习分类容易过拟合。通常，图像越少，越难训练更深更大的 CNN。如果在 A 中先训练网

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

络，然后移除 softmax 层，我们可以重建对 B 进行分类的 softmax 层。除了最后一个 softmax 层之外，所有层都会被保留下来，希望两个数据集共享相同的功能。

我们通过 CNN 的第一层来理解上述过程。在第 4 章中我们知道第一层学习到的是边缘和点检测器。而在第 1 章中，我们还看到如果对使用形状进行分类感兴趣，一些检测器如方向梯度直方图（HOG），可以被用作大多数数据集的特征检测器。一旦在数据集上训练了 CNN，我们就可以认为网络包含两个不同的组件：特征提取器和分类器。

图 5-1 是一个完全训练的 CNN，其中最后一层之前的 CNN 可看作特征提取器，最后一层的输出激活是特征向量，softmax 层是分类器。我们可以把不包括最后一层的网络看成与 HOG 非常相似的特征提取器。不同的是，我们不是人工构建，而是使用数据集本身，或更大的数据集学到该数据集的理想特征。进入分类层之前的特征向量是图像的特征表示，与我们在第 2 章中创建的员工表示类似。

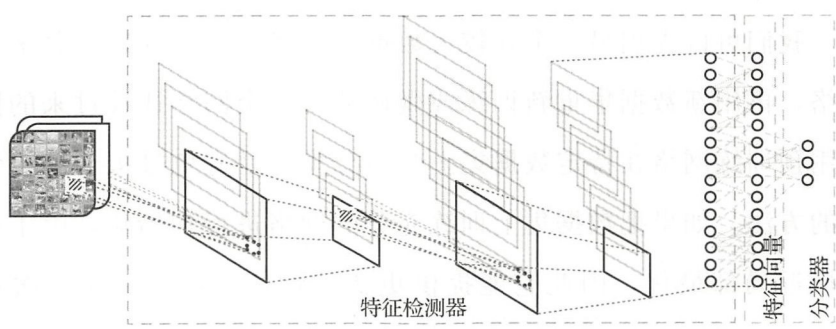


图 5-1 CNN 作为特征检测器，后面是分类器

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

最后一层没有什么，就是第 3 章中介绍的简单 Logistic 回归。上述情形实际上是不存在的，因为我们无法将网络与分类器完全分离。协同适应（coadaptation）在这些网络的学习中起着重要作用，因此学习权重和分类器同样重要，这样的结论对我们很有帮助。

从这种观点来看，如果数据集中的图像具有相似的性质（如局部图像属性和结构），那么从一个数据集学习的特征能轻易地传递到另一个数据集。一个更基本的观点认为，许多图像数据集的基本构建块几乎相同。将网络作为通用特征提取器的直觉是，从一个数据集学习的特征一定程度上在其他数据集中也适用。但是这种特性会随着网络的深度增加而削弱，因为越深的网络层越特定于数据集。这与创建通用特征的观点相似，如 HOG。HOG 期望产生区分图像形状的特征空间，而不限定数据集及分类任务。如果 HOG 是手工设计，且能在许多数据集通用，在 ImageNet 上表现良好，那么在 ImageNet 上训练且性能良好的网络，在其他数据集上同样具有好的泛化能力。

为进一步确保这些特征很好地适用，我们没有直接将它作为标准特征提取器。我们可以先用另一个在较大数据集训练的特征提取部分来初始化新的网络，再用新数据集重新训练或微调从另一个网络迁移过来的网络权重。利用预训练网络在特定数据集上学习分类，是避免过拟合和更快训练强有力的方法。如果把数据集上训练充分的网络进行初始化，由于在开始时已经接近局部最优，因此它能提供更快的训练。这类似于一次正则化操作。

特征提取器的目的是将图像映射到可区分空间。许多这样的网络已经

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

使用大型数据集（如 ImageNet）进行训练并提供公开下载，以便在特定数据集上微调（Russakovsky 等人，2015）。前几章中讨论的网络，如 AlexNet、GoogLeNet 和 VGG-19 都是现成的、可下载的和预训练的网络（Krizhevsky 等人，2012；Simonyan 和 Zisserman，2014；Soekhoe 等人，2016；Szegedy 等人，2015）。

使用预训练的网络会引发两个重要问题：

1. 如何选择构建预训练网络的数据集来学习特征？
2. 这些特征是否可传递给另一个数据集，如果可以，能传递得多好？

第一个问题的回答是数据集的神经通用性（Venkatesan 等人，2015，2016）。第二个问题的回答是神经特征的可传递性（Yosinski 等人，2014）。

5.1.1 通用性和可传递性

不同的数据集学习的滤波器不同。图 5-2 展示了一些识别手写字符的数据集。这些数据集中，我们期望 MNIST-旋转（MNIST-rotated）包含比 MNIST 更通用的特征。由于变换，MNIST-旋转包含更多的结构，因此 MNIST-旋转需要学习更复杂的滤波器。例如，MNIST-旋转训练的网络需要更多用于检测定向和边缘的通用滤波器。

如果数据集相似，则从不同数据集学习的滤波器也是相似的。网络的浅层滤波器，通常表示某些原子结构的检测器。原子结构是指 CNN 滤波器的一些形式，这些形式适应了所学数据集的熵。MNIST-旋转具有更大的熵，因此原子结构更多。熵高的数据集具有更多样复杂的原子结构。在

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

第 4 章中，MNIST 的第一层产生了简单的边缘检测器，MNIST-旋转会产生更复杂的特征，而由于 VGG 在更大更复杂的数据集进行训练，因此在角、边缘、点等上有复杂的检测器。一般来说，在 MNIST 上训练，VGG 网络比 LeNet 具有更多的原子结构，另外，尽管 LeNet 的原子结构数不如 VGG，但也已足够。使用 LeNet 处理 MNIST 数据，而使用 VGG 处理 ImageNet 数据明显要复杂得多（见图 5-3）。原子结构是网络训练具体的数据集而产生的，而不是通用结构。

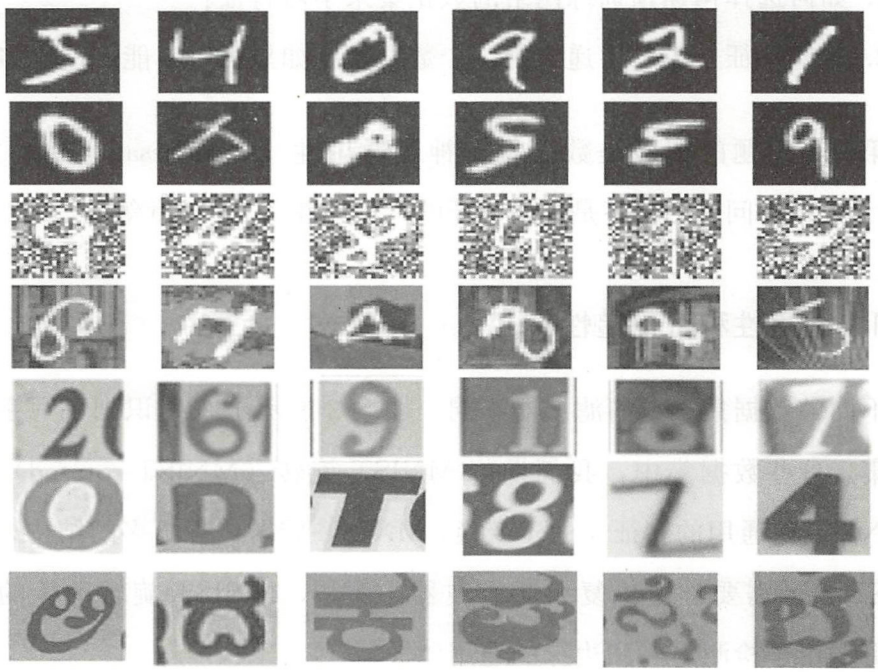


图 5-2　手写字符数据集。自上到下：MNIST（LeCun 等人，1998），MNIST－旋转（Larochelle 等人，2007），MNIST 随机背景，MNIST－旋转＋背景，谷歌街景房号（Netzer 等人，2011），Char74K 英文字符（de Campos 等人，2009），Char74K 坎纳达语字符。参考网站 convolution.network 查看彩色图

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

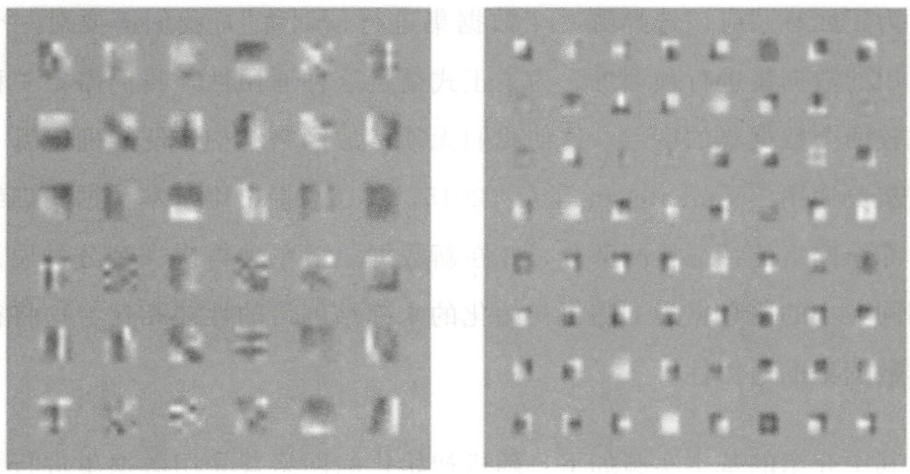


图 5-3 左侧是 Caltech 101 学得的滤波器，右侧是 VGG 第一层的滤波器。参考网站 convolution. network 查看彩色图

做以下思维实验：假设所有的原子结构构成一些空间 S，并包含三个数据集 $D=\{D_1, D_2, D_3\}$，$D\in S$ 的原子结构的集合。假设数据集的原子结构所占空间如图 5-4所示。显然，D_1 相对于 D_2 和 D_3，是更通用的数据集。D_1 包括 D_2 和 D_3 原子结构的绝大部分，但后两者不然。另外，此思维实验仅用于说明，在现实环境中不起作用。

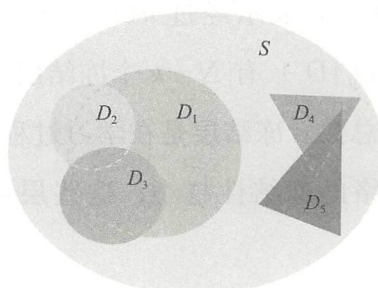


图 5-4 说明数据集通用性的思维实验。S 表示所有可能的原子结构空间，$D_1\sim D_5$ 分别表示不同数据集的原子结构所占空间

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

为了回答“应该选择哪一个数据集进行预训练”，我们需要对一个足够通用的数据集进行预训练。为了正式衡量这种通用性，我们需要一种能够比较两个数据集的度量。这使我们无须考虑网络结构，便能确定哪一个数据集更通用。Venkatesan 等人（2015，2016）提出用基本数据集训练网络，冻结网络的特征检测部分，并在新数据集上单独再次训练分类层。与在相同大小的原数据集上随机初始化的未冻结网络的性能相比，这将得到基本数据集的通用性。

其中介绍的基本实验如下：随机初始化的网络首先用数据集进行预训练。网络的该状态记为 $n(D_i|r)$，其中 D_i 表示我们正在初始化的数据集，r 为网络的原始随机设置；再根据图 5-5 的设置再训练 $n(D_i|r)$。

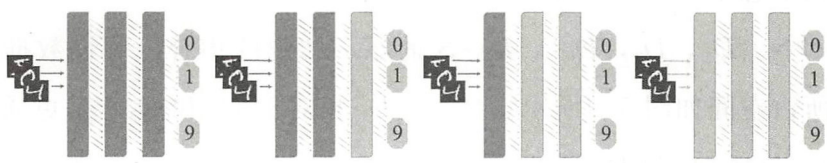


图 5-5　通用度量预测时的冻结和非冻结层

网络状态为 $n_k(D_j|D_i)$，表示经过 $n(D_i|r)$ 预训练后，用数据集 D_j 学习 k 个层的状态。$n_k(D_j|D_i)$ 有 $N-k$ 个冻结层（不可学习）来保留 D_i 的结果，其中 N 是层的总数。冻结层是在学习过程中冻结的层，我们可学习的层越多，意味着网络的冻结性越小；这些层可以是卷积及全连接神经层。

层学习包括两个方面，一方面是学习自己的内容，另一方面是学习前面层的非冻结层。一些层冻结化后，以此作为预先完成的变换。注意 $n_k(D_j|D_i)$

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

的性能提升不仅因为是新的第 $k+1$ 层被学习了，而且所有的 $k+1$ 层的组合被学习了。

图 5-5 演示了实验的设置。softmax 层在各种冻结设置中保持不变，即每次都进行随机初始化。这是由于不同数据集的标签数量或标签空间不尽相同。层的解冻从最后的层开始，未冻结的层不能进入冻结层。这是由于未冻结层学习新滤波器从而得到图像新特征的同时，后一个冻结层将无法适应变换。当未冻结层有两层时，这两层可以共同适应，并通过未冻结的路径进入未冻结的分类层。

如果 $n(D_j|r)$ 的泛化性能记为 $\Psi(D_j|r)$，而 $n_k(D_j|D_i)$ 的泛化性能记为 $\Psi_k(D_j|D_i)$，那么 D_i 对 D_j（在 k 层）的数据集通用性为：

$$g_k(D_i,D_j)=\frac{\Psi_k(D_j|D_i)}{\Psi(D_j|r)} \tag{5.1}$$

其中，$g_k(D_i, D_j)$ 表示 D_j 的泛化性能，它使用了 $N-k$ 个冻结层和具有新信息的 k 个层；k 个层由 D_i 的旧特征和对 D_j 的学习融合而来。使用该通用性度量，我们可以声明哪些数据集更适合进行网络预训练，同时保留新数据集的目标。研究表明，仅某个数据集的部分类就足够通用，因此不必对整个数据集作预训练。当某些类的样本数量少，而其他类数量很多时，类内通用性有利于我们用部分数据集训练模型，之后再用整个数据集进行再训练。

为了回答第二个可传递性问题，Yosinski 等人（2014）做了一项研究。实验设置与上述类似，但使用相同数据集的两个部分来执行。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Yosinski 不完全冻结和解冻这些层，而是保持所有层解冻并初始化新层。实验使用大数据集 ImageNet，将其分成两部分，分别用于训练和再训练。他们研究每层的可传递性和协同适应能力在学习过程中的作用。将 ImageNet 数据集的第一部分用来学习整个网络，再初始化部分网络，但保留其他部分；再用数据集的第二部分再训练网络。更有意思的是记忆能力分析。再训练第二部分后，返回并再次测试第一部分数据，以确定网络“记得”多少特征空间，即在新的特征空间中训练旧的分类器。这给出了关于跨（部分）数据集的特征质量好坏的思考。更多分析可阅读原文（Yosinski 等人，2014）。该文提出可传递性度量标准，展示了在相同数据集的两个部分训练的网络，它们的特征传递引起负面影响的问题。同时研究了层的协同适应及其在可传递性中的作用。他们发现，即使用其他任务构建的网络权重来初始化，也会比随机初始化效果好。关于通用性的文章是可传递性的后续，它证实了结论，并进一步阐明了其特性。

5.1.2　利用预训练网络的模型压缩

在上节中，我们研究了在通用数据集上训练的网络可作为特定数据集的初始化及微调。在此过程中，我们保留了网络的结构和优化器。因此，微调需要有与最初训练网络的机器类似的计算资源。随着深入学习在日常消费品中的普及和无处不在，我们有理由希望找到一种解决方案，使得产生类似的分类性能，同时需要较少的资源。

虽然能在 GPU 及其集群上训练大而深的网络，但它们的内存占用太大，无法适应手机和其他小型消费设备。研究表明，大型网络的大部分神经元作用不大（HasanPour 等人，2016；Li 等人，2016a，b；Nguyen 等

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

人，2015；Wang 和 Liang，2016）。大多数神经元或是功能上相似，或是对熵的表示无贡献；因此，可以通过压缩网络模型以便提高可移植性（Bucilua 等人，2006）。有更多理由解释为什么人们倾向于更小或中型网络；尽管大型的预训练网络的解决方案更好，但其由于计算量大，深度通常超过 20 层而看不到终点。一种更快地训练大型网络的方法是学习一个小网络，并使用如 Net2Net 转移学习技术将网络构建成更大的网络（Chen 等人，2015）。Net2Net 使用确定性方法将网络从小到大扩展，但转换是以函数保留的方式进行，从而加宽或加深网络并保留了从一层到另一层的函数映射。这样，一旦建立网络我们就能作再训练以获得更好的性能。鉴于优化和反向传播技术的理论基础，这种情况下可能发生的最坏情况是性能不再提高并保持稳定。但即使有了这些技术，这些网络的计算需求也不容易满足。大部分网络即使在简单的前馈模式下，也需要最先进的 GPU 才能工作。在较小的计算形式因素上使用预训练网络是不切实际的，因此需要学习较小的网络体系结构（Srivastava 等人，2015）。现在的难题是较小的网络体系结构无法产生强大的表示。

Hinton 等人（2015）提出了一种进行压缩并且保留表示的方法。虽然利用修剪和脑损伤实现压缩的原理是一个重要的研究领域，并且有了巨大进步，但本章我们只关注 Hintonian 压缩技术（Han 等人，2015；LeCun 等人，1989；Ullrich 等人，2017）。Hinton 发现反向传播及其学习只是网络实际学习的一小部分。尽管很诧异，但网络似乎也在学习类之间的关系。Hinton 进一步证明，虽然网络在对特定目标进行训练，但学习的趋势不受既定目标控制。使用网络在表示中学习类之间关系的思想，它能够为一个小得多的网络创建更好的目标。由于监督的熵更大，网络需要用较少

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

的神经元来泛化。本节用这一思想讨论模型压缩。

尽管我们假设类之间相互独立，每个类有自己的损失，但这不完全正确，如 MNIST 中，有些“2”看上去像“7”，有些“9”看上去像“6”。尽管没有明确监督，但网络本身会在类之间形成关系和比较。如图 5-6 所示，硬目标为标记，此时，我们正在学习一只狗，代表狗的标记位为 1，其他为 0。这就是我们所期望的 softmax，狗的概率接近 1，其余的概率接近 0。假设有一个大型网络或一组网络已经在这个数据集上学习，一旦我们通过它前馈狗的图像，它就产生如图 5-6 所示的 softmax 集成的输出。在这个 softmax 集成输出中，显而易见的是，狗的概率非常高，猫的概率次之，但仅是狗的 10%；尽管可能在浮点精度之内，但其他标签的概率随后更低。另外，第二大概率毕竟不是一个坏选择；在其他选择中，猫看上去最接近狗。第 4 章中 AlexNet 的输出也有类似效果。

奶牛	狗	猫	…	标签类别
0	1	0	…	硬标签
0.000006	.9	0.99991	…	集成
0.05	0.3	0.2	…	软标签

图 5-6　软化输出概率

Hinton 等人对这些 softmax 给出新形式；尽管类之间的相似性没有作为监督提供出来，但网络确实在隐含地学到了这种特性。这种情况下，尽管我们没有为这个事实创建任何特殊标签，但是网络发现猫看起来像狗的可能性很小。从第 1 章讨论的特征空间数据分布的角度看，可以思考：网络学习以这样的方式区分类的表示——相关的类更接近甚至部分重叠。我们未特别标注但网络本身学得的这些信息，我们称其为暗知识。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

本文用温度 softmax 代替常规的 softmax，定义为：

$$p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)} \tag{5.2}$$

其中 T 为温度，这种变换使整体预测更平滑，差异更显著。$T=1$ 为正常的 softmax，更高的温度会分散概率。温度越高，两个类别之间的 softmax 差异就越小。通过使用这个变换，我们可以清楚地看到网络猜测的内容，作为它的第二个和以后的选项。图 5-6 显示了这种通过温度升高的 softmax 软化标签的情况。

该温度 softmax 输出提供了关于类之间存在何种连接的观点，考虑上节的微调，我们希望用更少的神经元或层来实现，同时具有相同的表示能力；为此需要比数据集本身拥有更强的监督。我们在第一个预处理网络中使用温度升高的 softmax 输出作为额外的监督，引导较小的网络及数据集的自身损失。

对于较小网络上的典型微调，我们使用硬标签上误差的反向传播。在这种情况下，如果将另一网络的硬标签反向传播的误差传播到温度升高的软标签上，那么我们能提供更多的监督。因为软目标训练难以产生强梯度，所以实际中会创建一个软目标和硬目标加权的误差函数。可以将其看作暗知识。假设 $p_T(i)$ 是第 i 批次输入到 p 的温度 softmax 输出，$y(i)$ 是真实标签，$c(i)$ 是 softmax 输出，$c_T(i)$ 是当前学习子网络的升温 softmax 输出。$\| c(i) - y(i) \|$ 是该批次温度 softmax 输出 $\| c_T(i) - p_T(i) \|$ 的差。新的反向传播损失为：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

$$e(w) = \| c(i) - y(i) \| + \alpha \| c_T(i) - p_T(i) \| \tag{5.3}$$

作者可以将 MNIST 数据集的暗知识从 1200-1200-10 网络转移到 800-800-10 网络，而仅有 0.02%的损失。对于后续结果，他们通过保留特定类的样本空间，仅使用暗信息转移进行学习。例如，如果在训练子网络时不使用类别 3 且只有类别 3 的软目标被传播，则网络对类别 3 仍能正确分类。

这种技术称为模型压缩，或者暗知识。暗知识转移也是一种强正则化，但与 L1 和 L2 正则化惩罚过大或过稀疏的权重不同，它来自另一个网络的暗知识，用来惩罚与先前网络不一致的输出。这可以作为子网络对父网络输出分布的建模，这表示训练较小的网络有潜力将图像映射到较大网络相同的标签空间，且保持有效。

目前，许多计算机视觉架构师宁可在计算资源受限的预训练网络进行微调，也不愿从头训练新的体系结构。然而，即使训练小数据集的中型网络也非常困难；训练一个深度网络，甚至是中等的深度网络都需要大量监督来避免权重爆炸。在这些情况下，暗知识转移有助于提供一些监督和规范。

5.1.3　Mentee 网络与 FitNet

Romero 等人对暗知识做了进一步研究，对软标签和中间层激活一起作反向传播（Romero 等人，2014）。这为增加更多监督提供了广阔前景。包括这种监督的网络称为 Mentee 网络，它如同导师网络提供监督。上文中作者最初希望制作一个更薄更深的父网络。类比于 CNN 和其他神经网络，深网络比宽网络更具效率。这一主张在 AlexNet（LeCun 等人，

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

2015）中已进行论证。如果我们有两个不同配置的网络，我们先使用暗知识转移来学习相同的标签映射，再思考转移 softmax 层及中间层。

Mentee 网络最初由 Romero 等人以 FitNet 的方式提出，用来学习更薄更深的网络（2014）。一些文献说明了这个观点（Chan 等人，2015；Gupta 等人，2015；Venkatesan 等人，2015，2016）。最近的论文将其作为好的初始值设定项（Tang 等人，2015）。另一些人使用该技术在不同类型网络之间传递知识，如递归神经网络（RNN）和 CNN（Geras 等人，2015；Shin 等人，2016）。Mentee 网络的基本思想是传播激活并使另一个网络学习相似能力的特征空间，即较小的网络可以对较大网络的特征空间及其中间层进行表示。

图 5-7 描述了学习该网络的连接。上面的网络是一个预训练的现成网络，下面的网络是一个随机初始化的不同网络。它们的前馈过程相同，并

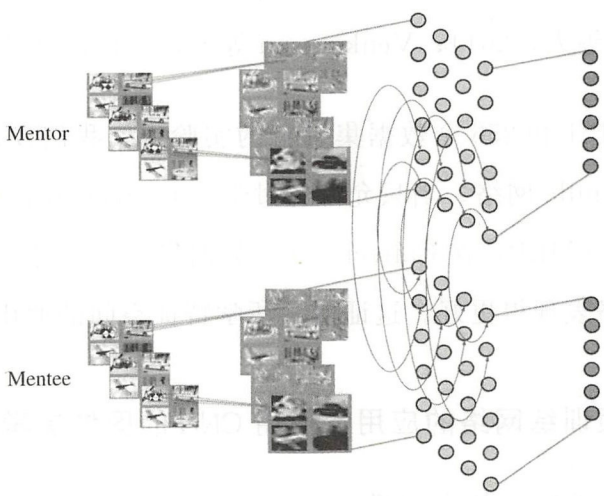


图 5-7 Mentor-mentee 连接

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

期望生成相同的 softmax 输出（使用暗知识转移）。为此，我们希望网络的某些层能产生类似的激活。网络上的这些连接仅在训练中使用，训练后规模较小的 Mentee 网络就能够工作；且反向传播仅在较小的 Mentee 网络中进行，更大的导师（Mentor）网络只进行正向传播。

此网络的常用权重更新如下所示：

$$w^{t+1} = w' - \eta\left[\alpha_t \frac{\partial}{\partial w} e(w) + \beta_t \sum_{\forall (l,j) \in B} \frac{\partial}{\partial w} \Psi(l,j) + \gamma_t \frac{\partial}{\partial w} \Psi(n,m)\right] \tag{5.4}$$

其中 n 和 m 是层的数量（包括网络中的 softmax 层），B 包括从导师到被监督 Mentee 的层对。方括号的第一项是正常的随机梯度下降误差的更新，第二项是导师权重更新，最后一项是暗知识转移权重更新。α、β 和 γ 的设置决定着不同学习的发生。研究表明，利用不同的学习速率，可以产生这些 Mentee 网络的几个不同特征，甚至可以直接学习导师网络的确切特征空间（Romero 等人，2014；Venkatesan 等人，2015，2016）。

在 Caltech101 和 256 个数据集进行的实验中，我们发现 Mentee 网络的性能优于 vannilla 网络。当只允许经过学习 Caltech 101 学到表示的分类器/多层感知器（MLP）在 Caltech－256 数据集上学习时，Mentee 网络也能够比独立网络表现得更好。这证明了所学特征空间的通用性。

5.1.4　使用预训练网络的应用：使用 CNN 的图像美学

将网络预先放置在某个图像分类任务上，研究它在不同任务上的应用。我们考虑比较图像的美学，这与分类有很大的不同（Chandakkar 等

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

人，2017）。

CNN 在“客观评价是可能的”的任务中取得了成功；然而，也有一些固有的主观任务，例如图像美学的评估。CNN 在这类任务中也表现出卓越的效果，并优于基于摄影文献中基于事实的手工特征的方法。CNN 可能已经发现了一些对图像至关重要但尚未定义的特性（Lu，2014，2015）。

最早由 Lu（2014）提出使用经过预训练的 AlexNet CNN 进行图像美学自动化评估。RAPID 包含一个双通道 CNN，即两个 AlexNet 捆绑在一起。一个 CNN 将整个图像作为输入，而另一个 CNN 将随机裁剪的区域作为输入。最后，组合这两个输入的特征向量（最后一层的表示），初始化新的回归器以产生图像的美学分数。

考虑到这项任务的主观性：一方面，人们可能会认为给每个图像分配绝对分数是困难的，因此容易出错。另一方面，比较两幅图像的审美品质可能更可靠。这也有实际应用。常见的是，用户有一组照片，用户只关心它们的排名，而不是单个照片的分数。另外，例如，对于基于美学和图像增强的图像检索，这种基于排名的方法是必要的。这就是 Chandakkar 等人（2017）提出的方法背后的动机。

Chandakkar 等人（2017）工作的主要思想如图 5-8 所示。它以一对图像作为输入，使用类似于 RAPID 方法的架构处理每个图像。两幅图像的特征向量分别称为 C_1 和 C_2，对它们作相应处理来产生表示图像之间的美学等级的分数。损耗函数采用 max（0，）的形式。当两幅图像的排名与事实不符时产生损失，否则损失为 0。因此，不需要对 AlexNet 的表示学习

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

系统进行过多训练，就能将其应用到一个完全不同的美学任务中。

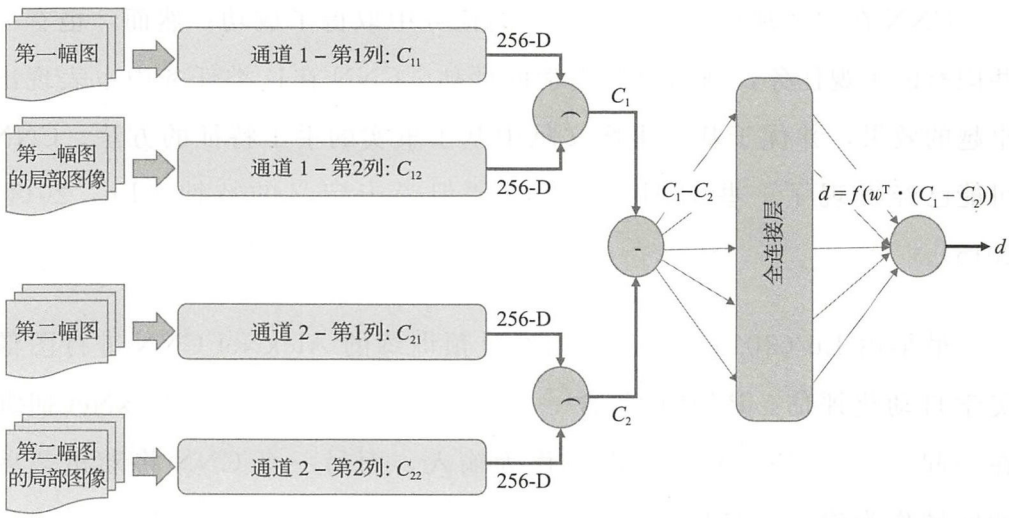


图 5-8　美学网络的架构；权重在 C_{11} 和 C_{21}、C_{12} 和 C_{22} 列之间共享；连接 C_{11} 和 C_{12} 获得的特征（用符号表示）以获得 C_1，连接 C_{21} 和 C_{22} 以获得 C_2；向量 C_1-C_2 穿过两个密集层，以获得比较两个图像美学的得分 d。f () 表示 ReLU 非线性。详情请参阅文献（Parag Chandakkar）

上述过程表明，即使 AlexNet 接受部分图像分类任务的训练，学习过程也可能使网络找出足够通用的特征，从而将其用于似乎完全不同的任务中。Hu 等人（2016）提出的 video2vec 框架是另一个很好的例子，它将预处理后的 CNN 与 RNN 结合使用，学习视频的语义时空嵌入空间，从而更好地支持动作识别等视频分析任务。

5.2　生成网络

到目前为止，我们研究的网络都试图使预测分类标签上分布的损失最

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

小化。这些通常被称为分类器网络。这些网络从高维空间（例如图像）的分布中采样数据，并将其映射到标签空间或特征表示的某个空间。还有一类网络是这些网络的必然结果，称为生成网络。生成网络从通常比数据空间小得多的潜在空间上的分布中采样数据，并生成模仿从高维空间的分布中提取的样本的样本。为了有利于计算机视觉，这些网络从潜在空间的分布中取样并产生图像。在本节中，我们研究两种最流行的生成网络模型。

5.2.1 自动编码器

自动编码器可能是不用于分类的 CNN 网络中最突出的案例。如图 5-9 所示的自动编码器是学习自我再生图像的镜像对称网络。自动编码器由编码器和解码器两部分组成。编码器创建图像的简洁而紧凑的表示，通常称为码字或潜在空间表示。解码器从这些潜在空间表示中重构图像。自动编码器可与字典学习或主成分分析（PCA）相比，因为它们创建的表示更接近数据的真实底层结构，而与分类标签无关，并且在较低（最常见）的维度空间中。通常，解码器不与编码器分离。通常解码器权重只是编码器权重的转置，因此解码器和编码器是对称的，并且共享权重。如果这些层在编码器上是卷积的，我们可以使用转置卷积或分数阶卷积。这些也（有些滥用术语）被称为反卷积。这是简单的卷积运算的梯度，这些卷积运算是对转置滤波器的卷积。对于最大池化，我们使用阶卷积来反池化或保存最大池化的位置进行反池化，这些被称为分数阶卷积；它们在卷积前做上采样。有人认为这类似于学习反池化操作的自定义上采样操作（Zeiler 等人，2010）。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

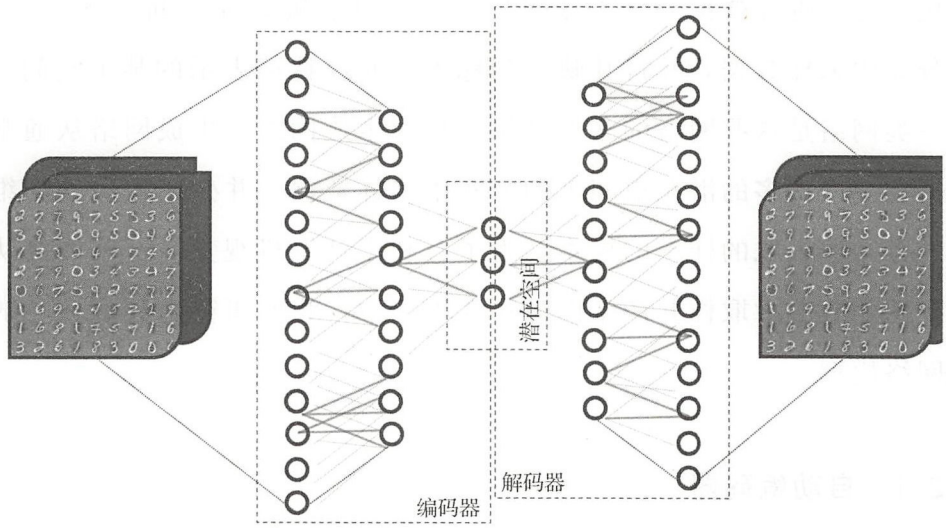


图 5-9　自动编码器

一个自动编码器有两个部分，编码器 $E(x)=c$，它接收 x，并产生图像 c 的码字表示；解码器 $D(c)=\hat{x}$，它接收码字 c，并产生图像 $\hat{x}$ 的重构。由于权重通常被绑定在编码器和解码器之间，所以我们需要学习 E 或 D 的权重。我们通过最小化 $\hat{x}$ 和 x 之间的重构误差来做到这一点。我们可以选择使用任何先前的误差来实现这一点，但是 RMSE 误差通常是优选的。图 5-10 显示了单层自动编码器的典型输入码字重构。用于创建图形的自动编码器的代码可在 Yann 教程中找到（附录 A）。

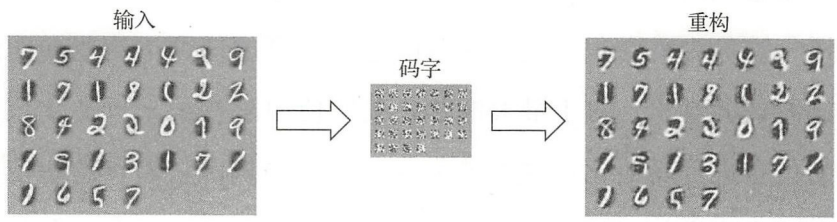


图 5-10　使用一个隐藏层的自动编码器的 MNIST 编码和解码过程

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

自动编码器有多种形式和特点。使用神经网络的自动编码器要么过采样要么欠采样。如果码字的维数小于输入的维数，则为欠采样自动编码器；否则，它是过采样的。过采样的自动编码器通常需要强正则化和稀疏性才能学习。由于图像是从嵌入的子空间重构的，因此不期望重构是完美的。虽然这可能是一个缺点，但自动编码器可以用来生成需要的非完美重构的图像。学习过采样的自动编码器是非常困难的。给输入添加噪声并期望重构无噪声图像是规范这些网络的一个强有力的方法，以便我们可以轻松地学习更大和更深的网络。

考虑去噪问题。在去噪中，我们要从图像中去除噪声。虽然自动编码器会使用图像中的一些结构来重构图像，但这些结构通常在嘈杂的图像中找不到。编码器仅编码图像中统计常见的特征。噪声是一个随机过程，因此不是数据集之间的共同结构。噪声图像和无噪图像的潜在空间几乎相同。因此，通过自动编码器前馈的噪声图像将可能生成干净的图像。去噪自动编码器是自动编码器最流行的实现方式之一（Vincent 等人，2008，2010）。结果发现，学习一组输入损坏的自动编码器优于学习一组干净的输入。这是因为学习特征形成了一组损坏的特征，与简单学习相比，这些特征有助于编码器学习更好更健壮的表示。损坏自动编码器的输入与我们先前研究的丢失正则化器非常相似。事实上，加到输入端的噪声之一是将随机位置零。Vincent 等人提出了第一种 dropout 算法。

自动编码器的编码器也可以用作分类网络的初始化。自动编码器是一种生成方法，因此它们学习的基本特征是纯的，不受类标签熵的影响。因此，从中学习到的特征只涉及图像中的结构。这样的网络，作为指导或完全预训练的网络初始化都有助于为分类网络建立强有力的初始化。各种表

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

现形式的自动编码器称为无监督学习，这是一个活跃的研究领域（Bengio，2012；Bengio 等人，2012；Erhan 等人，2010 年；Goodfellow 等人，2014 年；Radford 等人，2015）。

5.2.2　生成对抗网络

自动编码器生成接近样本平均值的样本，它使用潜在表征来表示。为了能够更好地进行高维采样，Goodfellow 等人（2014）设计了一种新的生成网络，称为对抗网络。它与以前的网络不同，它的目标不在网络的外部，而是在机器内部。对抗性生成学习涉及两个网络，其中一个是生成器，即前一节讨论的解码器。生成器从随机潜在分布中采样并生成图像。第二个是一个分类器，它试图预测输入图像是由生成器网络生成的还是从数据集采样的。如果数据从现实世界中采样，则在分类器的最后一层训练 softmax 层，以分类的方式学习特征。

GAN 的工作方式如下。考虑按 p_{data} 分布的数据，数据形式未知。尽管我们假设 p_{data} 是某种形式 p_{model} 的参数化分布。在极大似然下，希望找到一组好的参数，这样我们就无法区分从 p_{model} 生成的样本和 p_{data}。对参数 w 进行以下极大似然估计：

$$\ddot{w} = \text{argmax}_w \, \mathbb{E}_{x \to p_{data}} \log p_{model}(x|w) \tag{5.5}$$

在 GAN 中，我们有两个网络，生成器 $G(z)$ 以随机向量 z 作为输入并转换为图像，分类器 $D(x)$ 得到输入图像 x 并产生一个概率，它由真实值 x 和预测值的运算得到。GAN 系统本质上是 G 和 D 间的博弈。D 通过调整自身权重，目的是试图把现实世界的数据和 $G(z)$ 产生的数据区分开

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

来。G 调整其权重来产生 $G(z)$，使得 $D(G(z))=1$，这里用 D 作为 $D(x)$ 或 $D(G(z))$ 产生的概率，稍微有些滥用这个符号。GAN 博弈有以下两个步骤。对于 D，

$$\min_D -\mathbb{E}_{x\to p_{data}(x)}[\log(D(x))]-\mathbb{E}_{z\to p_z(z)}[\log(1-D(G(z)))] \quad (5.6)$$

其中 D 试图最小化 $D(G(z))$和最大化 $D(x)$，表示它正在学习区分假、真图像。对于 G，

$$\min_G -\mathbb{E}_{z\to p_z(z)}[\log(D(G(z)))] \quad (5.7)$$

其中 G 学习产生 $G(z)$，使 D 预测一个较大的 $D(G(z))$ 值。注意，在优化公式 5.6 时，仅更新 D 的权重，而在优化公式 5.7 时，仅更新 G 的权重。这意味着 G 从不观察任何图像，仍然学习生成图像。图 5-11 显示了一些生成的图像。

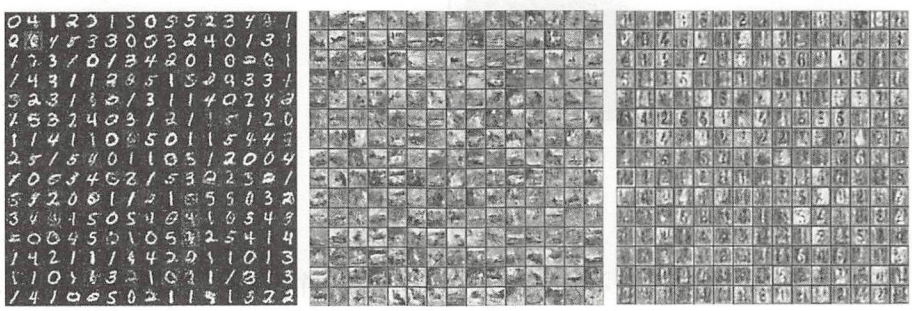


图 5-11 MNIST、CIFAR 10 和 SVHN 使用 0— 5 类的一些样本，使用 LeNet 大小的 G 和 D；它们以灰度级生成，这些图像的彩色版本可在卷积的书籍网页上获得。此外，生成这些图像的代码也可以在 yann 教程中找到

这些网络以这种最小最大博弈的方式同时训练，一些证据表明，它能

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

达到平衡状态；分类者无法预测数据是来自真实世界还是生成器（Goodfellow 等人，2014）。这使学习更快收敛。收敛时，{D} 总是产生 0.5 的值，而不管输入是从哪里采样的。图 5-12 显示了典型生成对抗网络（GAN）的体系结构。

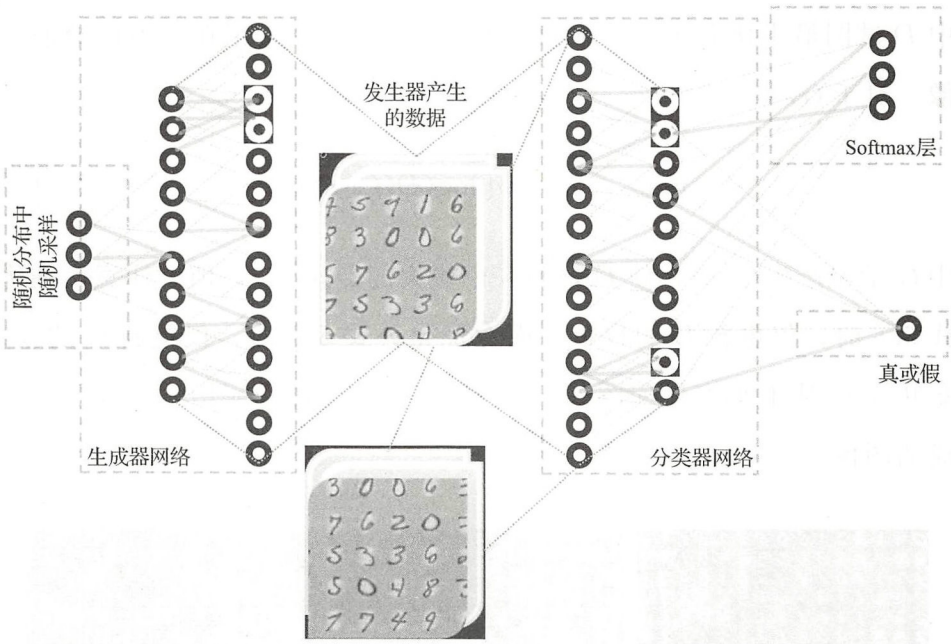


图 5-12　通用对抗网络

GAN 的训练通常从分类器开始，甚至可以首先对分类器初步训练，以便使用 softmax 层从数据集中识别样本。一旦分类器就绪，我们就输入未经训练的生成器产生的样本。分类器产生误差来预测图像是来自数据集还是生成器。随着学习的进行，生成器网络学习生成越来越接近原始数据的样本，直到生成与数据不可区分的点（至少对于分类器网络）。为了产生更好的图像，我们使用转置的卷积和反卷积 GAN 进行扩展（Radford 等人，2015）。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

GAN 的一个问题是生成网络具有陷入相同模式的趋势，即生成器不会捕获数据集的多样性，而只锁定一种类型的数据。在图 5-11 中，我们已经注意到一些类比其他类生产得更频繁。考虑图 5-13 中的情况，带有点的圆是原始数据集的分布。GAN 的任务是产生看起来是从这个原始数据集采样的数据。训练有素的 GAN 似乎可以从正方形填充区域表示的分布中采样数据。这种 GAN 将能够产生所有模式或类型的样本。停留在一种模式下的 GAN 将从单峰（或几种模式）分布中产生样本。这将导致一次又一次地产生相同或至少相似的数据。解决这一问题的一种方法是使用模式正则化 GAN（MRGAN）（Che 等人，2016）。在 MRGAN 中，作者介绍了稳定 GAN 模型训练目标的新方法。他们证明，这种正则化器有助于将概率跨域分布在生成数据的多种分布模式中。

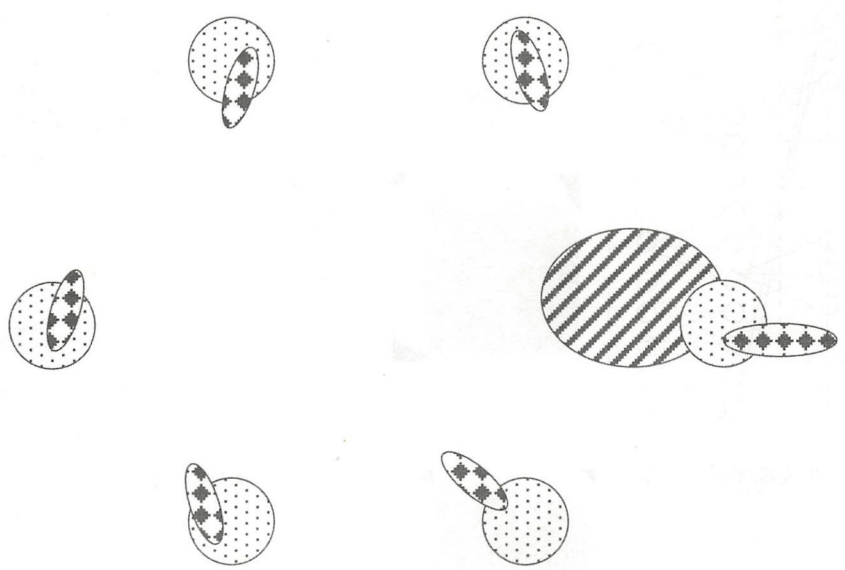


图 5-13 GAN 陷入一种模式：用点填充的五个圆是数据集的原始分布，具有相同填充的椭圆表示一个生成器的分布。斜线填充的生成器陷入在一种模式中，而正方形填充的生成器尽管不完美，但它是一个很好的生成器。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

GAN 最近增加了对这些生成器的可解释性。这是通过使用合理表示和 InfoGAN（Chen 等人，2016）来实现的。InfoGAN 试图最大化潜在变量的小的子集和生成器之间的相互信息。生成器不仅从随机分布中采样，还从描述学习中的样本的一些属性的潜在空间中采样。图 5-14 描述了这种体系结构。这允许网络梯度最大化生成器变量的信息。在最简单的可能形式中，生成器与分类器相反，它生成需要的类的样本意味着生成器是有区别地训练的。几个概念验证实验的证明在这篇文章（Chen 等人，2016）中可以找到。自从有了 GANs 的原始公式以来，已经迭代了多种 GAN。最小二乘 GAN 和 Wasserstein GAN 是较受欢迎的版本（Arjovsky 等人，2017；Mao 等人，2016）。

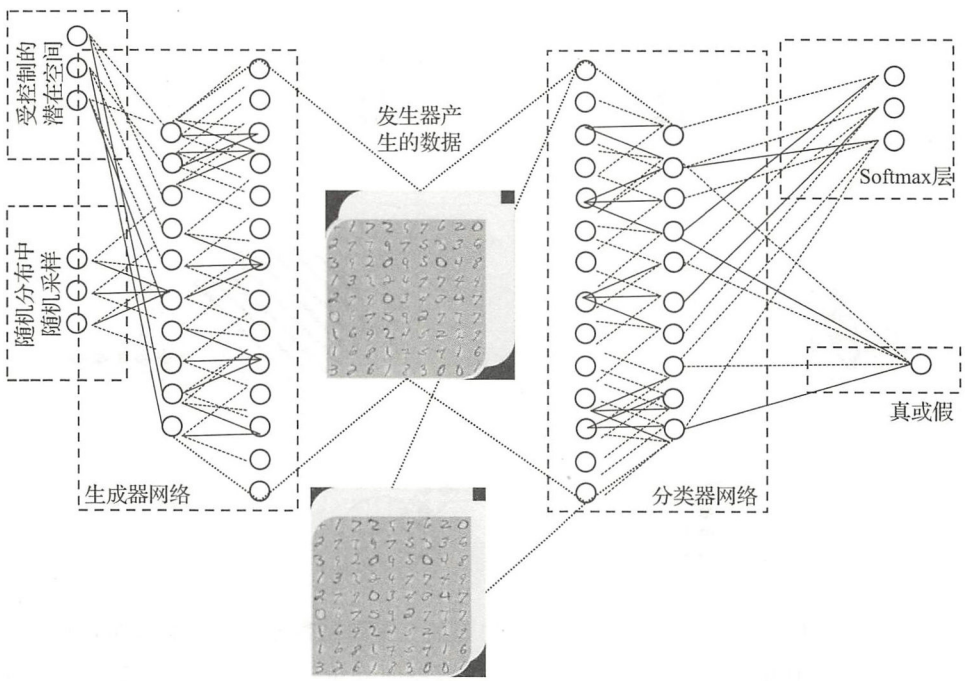


图 5-14　InfoGAN 结构

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

GAN 的一个优势是，不直接对本义的数据编码或压缩，而从学到的分布模型中取样本。即 GAN 的生成器是数据流的来源。这对许多应用非常有用，如增量学习（Venkatesan 等人，2017）。GAN 是神经网络的新飞跃，也是一个潜在宝库。

本章小结

本章研究了 CNN 的几种现代和新颖的用法，每种用法都为我们提供了关于如何使用深度网络的新视角。深度学习每天都以更有创意的方式来解决许多问题。使用深度学习技术开发的产品涉及面很广。从图像搜索和语音识别等消费品到严峻的医疗应用，深度学习已经成为现代技术中无处不在的组成部分。虽然有广泛的应用，但我们选择了简单的研究，让读者对这些网络的使用和应用有一个新的认识。

我们注意到，其中大部分可以通过使用不同的损失函数和激活来实现。我们讨论了图像数据集为何具有一些类似或一般的属性，这些属性有助于决定在训练时使用哪个数据集。我们研究了深度学习的可传递性，发现在一个数据集上学习的一些网络可以用来从另一个数据集学习。网络学到的东西远远超过了明确要求学习的东西。我们利用该点，试图用该类的样本学习类分布。我们将这一点扩展到简单的暗知识转移及让一个网络指导另一个网络。

本章还讨论了生成器学习，从中间层（码字层）重构图像。这种不完美的重构在去噪等情况下是有用的。CNN 充满了潜力，这些网络更新的和新颖的用途每天都在发生并帮助人类解决各种问题。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

参考文献

Arjovsky, Martin, Chintala, Soumith, and Bottou, L'eon. 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875.

Bengio, Yoshua. 2012. Deep learning of representations for unsupervised and transfer learning. *ICML Unsupervised and Transfer Learning* 27: 17–36.

Bengio, Yoshua, Courville, Aaron C, and Vincent, Pascal. 2012. *Unsupervised feature learning and deep learning: A review and new perspectives.* CoRR, abs/1206.5538, 1.

Buciluǎ, Cristian, Caruana, Rich, and Niculescu-Mizil, Alexandru. 2006. Model compression. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 535–541, San Fransisco, California. ACM.

Chan, William, Ke, Nan Rosemary, and Lane, Ian. 2015. *Transferring knowledge from a RNN to a DNN.* arXiv preprint arXiv:1504.01483.

Chandakkar, Parag, Gattupalli, Vijetha, and Li, Baoxin. 2017. A computational approach to relative aesthetics. arXiv preprint arXiv:1704.01248.

Che, Tong, Li, Yanran, Paul Jacob, Athul et al. 2016. *Mode regularized generative adversarial networks.* arXiv preprint arXiv:1612.02136.

Chen, Tianqi, Goodfellow, Ian, and Shlens, Jonathon. 2015. Net2Net: Accelerating learning via knowledge transfer. arXiv preprint arXiv:1511.05641.

Chen, Xi, Duan, Yan, Houthooft, Rein et al. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, pp. 2172–2180, Barcelona, Spain. NIPS.

de Campos, TE, Babu, BR, and Varma, M. 2009. Character recognition in natural images. *Proceedings of the International Conference on Computer Vision Theory and Applications,* Lisbon, Portugal.

Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron et al. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11: 625–660.

Everingham, Mark, Van Gool, Luc, Williams, Christopher KI et al. n.d. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) results.* Retrieved from PASCAL VOC: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

Fei-Fei, Li, Fergus, Rob, and Perona, Pietro. 2006. One-shot learning of object categories. *IEEE Trans. Pattern Recognition and Machine Intelligence* (IEEE) 28: 594–611.

Geras, Krzysztof J, Mohamed, Abdel-rahman, Caruana, Rich et al. 2015. *Compressing LSTMs into CNNs.* arXiv preprint arXiv:1511.06433.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi et al. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pp. 2672–2680.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Gupta, Saurabh, Hoffman, Judy, and Malik, Jitendra. 2015. *Cross modal distillation for supervision transfer*. arXiv preprint arXiv:1507.00448.

Han, Song, Mao, Huizi and Dally, William. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.

HasanPour, Seyyed Hossein, Rouhani, Mohammad, Vahidi, Javad et al. 2016. *Lets keep it simple: Using simple architectures to outperform deeper architectures*. arXiv preprint arXiv:1608.06037.

Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeff. 2015. *Distilling the knowledge in a neural network*. arXiv preprint arXiv:1503.02531.

Horn, Berthold KP and Schunck, Brian G. 1981. Determining optical flow. *Artificial Intelligence* (Elsevier) 17: 185–203.

Hu, Sheng-heng, Li, Yikang, and Li, Baoxin. 2016. Video2Vec: Learning semantic spatial-temporal embeddings for video representation. In *International Conference on Pattern Recognition*, Cancun, Mexico.

Karpathy, Andrej and Fei-Fei, Li. 2015. Deep visual-semantic alignments for generating image descriptions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, Santiago, Chile. IEEE.

Krizhevsky, Alex, Sutskever, Ilya, and Geoffrey, Hinton E. 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, Harrahs and Harveys, Lake Tahoe: NIPS, pp. 1097–1105.

Larochelle, Hugo, Erhan, Dumitru, Courville, Aaron et al. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. *Proceedings of the 24th International Conference on Machine Learning*, pp. 473–480, Corvallis, Oregon. ACM.

LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. 2015. Deep learning. *Nature* 52: 436–444.

LeCun, Yann, Bottou, Leon, Bengio, Yoshua et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.

LeCun, Yann, Denker, John S, Solla, Sara A, Howard, Richard E and Jackel, Lawrence D. 1989. Optimal brain damage. *NIPs* 2: 598–605.

Li, Hao, Kadav, Asim, Durdanovic, Igor et al. 2016a. *Pruning filters for efficient ConvNets*. arXiv preprint arXiv:1608.08710.

Li, Yikang, Hu, Sheng-hung, and Li, Baoxin. 2016b. Recognizing unseen actions in a domain-adapted embedding space. *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 4195–4199, Phoenix, Arizona. IEEE.

Lu, Xin, Lin, Zhe, Jin, Hailin et al. RAPID: Rating pictorial aesthetics using deep learning. *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 457–466. Orlando, FL, ACM.

Lu, Xin, Lin, Zhe, Shen, Xiaohui et al. 2015. Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 990–998.

Mao, Xudong, Li, Qing, Xie, Haoran, Lau, Raymond YK, Wang, Zhen, Smolley, Stephen Paul. 2016. Least squares generative adversarial networks. arXiv

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

preprint ArXiv:1611.04076.

Netzer, Yuval, Wang, Tao, Coates, Adam et al. 2011. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupevised Feaure Learning*.

Nguyen, Anh, Yosinski, Jason, and Clune, Jeff. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, Boston, MA. IEEE.

Radford, Alec, Metz, Luke, and Chintala, Soumith. 2015. *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv preprint arXiv:1511.06434.

Romero, Adriana, Ballas, Nicolas, Kahou, Samira et al. 2014. *FitNets: Hints for thin deep nets*. arXiv preprint arXiv:1412.6550.

Russakovsky, Olga, Deng, Jia, Su, Hao et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115: 211–252.

Shin, Sungho, Hwang, Kyuyeon, and Sung, Wonyong. 2016. *Generative transfer learning between recurrent neural networks*. arXiv preprint arXiv:1608.04077.

Simonyan, Karen and Zisserman, Andrew. 2014. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

Soekhoe, Deepak, van der Putten, Peter, and Plaat, Aske. 2016. On the impact of data set size in transfer learning using deep neural networks. *International Symposium on Intelligent Data Analysis,* pp. 50–60.

Srivastava, Rupesh K., Greff, Klaus, and Schmidhuber, Jurgen. 2015. Training very deep networks. *Advances in Neural Information Processing Systems*, pp. 2377–2385. NIPS.

Szegedy, Christian, Liu, Wei, Jia, Yangqing et al. 2015. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA. IEEE.

Tang, Zhiyuan, Wang, Dong, Pan, Yiqiao et al. 2015. *Knowledge transfer pre-training*. arXiv preprint arXiv:1506.02256.

Ullrich, Karen, Meeds, Edward and Welling, Max. 2017. Soft weight-sharing for neural network compression. arXiv preprint arXiv:1702.04008.

Venkatesan, Ragav, Chandakkar, Parag, and Li, Baoxin. 2015. Simpler non-parametric methods provide as good or better results to multiple-instance learning. *Proceedings of the IEEE International Conference on Computer Vision*, Sanitago, Chile: IEEE, pp. 2605–2613.

Venkatesan, Ragav, Gattupalli, Vijetha, and Li, Baoxin. 2016. On the generality of neural image features. *IEEE International Conference on Image Processing*. Phoenix: IEEE.

Venkatesan, Ragav, Venkateswara, Hemanth, Panchanathan, Sethuraman, and Li, Baoxin. 2017. A strategy for an uncompromising incremental learner. arXiv preprint arXiv:1705.00744.

Vincent, Pascal, Larochelle, Hugo, Bengio, Yoshua et al. 2008. Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103,

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Helsinki, Finland. ACM.

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle et al. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11: 3371–3408.

Wang, Xing and Liang, Jie. 2016. *Scalable compression of deep neural networks*. arXiv preprint arXiv:1608.07365.

Wasserman, Philip D. 1993. *Advanced Methods in Neural Computing*. New York, NY: John Wiley & Sons.

Yosinski, Jason, Clune, Jeff, Bengio, Yoshua et al. 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, pp. 3320–3328.

Zeiler, Matthew D, Krishnan, Dilip, Taylor, Graham W et al. 2010. Deconvolutional networks. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference*, pp. 2528–2535, San Fransisco, CA IEEE.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

附录 A

Convolutional Neural Networks in Visual Computing

Yann

我们正在启动一个工具箱来构建和学习卷积神经网络和其他在本书中研究的网络。工具箱被称为另一个神经网络（Yann）工具箱，托管在 http://www.yann.网络。YANN 工具箱是建立在 Python 2.7 和 Theano 框架之上的，未来计划能够同时使用 Theano 和 TensorFlow 框架（Bergstra 等人，2010；Abadi 等人，2016）。安装和设置 Yann 非常简单。需要做的仅是在 Linux 上运行以下命令，安装程序的简单版本：

```
>> pip install git+git://github.com/ragavvenkatesan/
  yann.git
```

Yann 处于早期阶段，目前正处于大规模发展阶段。虽然需要进行一些更改，但本章只介绍工具箱的核心，即使实际工具箱可以持续更新，它也应该保持有效。如果确实有变化，每个版本的版本说明及其 API 文档都会反映出来。这个附录对应于版本 1.0rc1，这是 Yann 版本 1 的发布候选。在工具箱文档中总是有本章的最新教程版本。虽然有更多的正式和有益的工具箱，有一个更大的用户基础，如 ConvNet、Lasagne、Keras、Blocks 和 Caffe（Lasagne，n.d.；Chollet，2015；Vedaldi 和 Lenc，2015；Jia 等人，2014），这个工具箱对于初学者来说学习起来非常简单和通用。因此，Yann 是这本书的完美补充。它也是一个很好的选择，用于运行预先训练的模型和构建复杂的 nonvanilla 体系结构，这是不容易用其他工具箱构建的。Yann 也

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

被用于深度学习计算机视觉课程（具体内容参见 course. convolution. network），这是从本书中开发的，并于 2017 春季在 ASU 教授。

Yann 的架构

Yann 工具箱的核心及其操作是围绕 yann. network. network 类构建的，该类存在于文件 yann/network. py 中。图 A-1 显示了 yann. network. network 类的组织。add _ 方法增加了一层或模块作为命名。网络类可以在使用 add _ 方法添加的各种连接和体系结构中保存多个层和模块。在为学习准备网络时，我们可能只需要某些模块和层。通过选择和建立网络的训练、测试和验证部分来准备网络的过程称为 cooking（编写）。

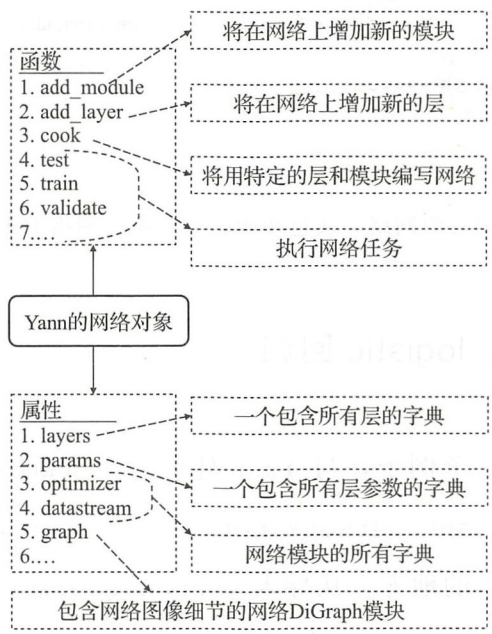


图 A-1　网络对象的结构

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

一旦编写完成，网络就可以通过调用 network 内的其他函数来进行训练和测试。该 network 类也存在其他一些属性，如 layers 和 params，前者是增加进该类的层的字典，而后者是所有参数的字典。所有的 layers 和 modules 包含一个 id 属性。通过该属性我们可以对 layers 和 modules 进行访问（如图 A-2 所示）。

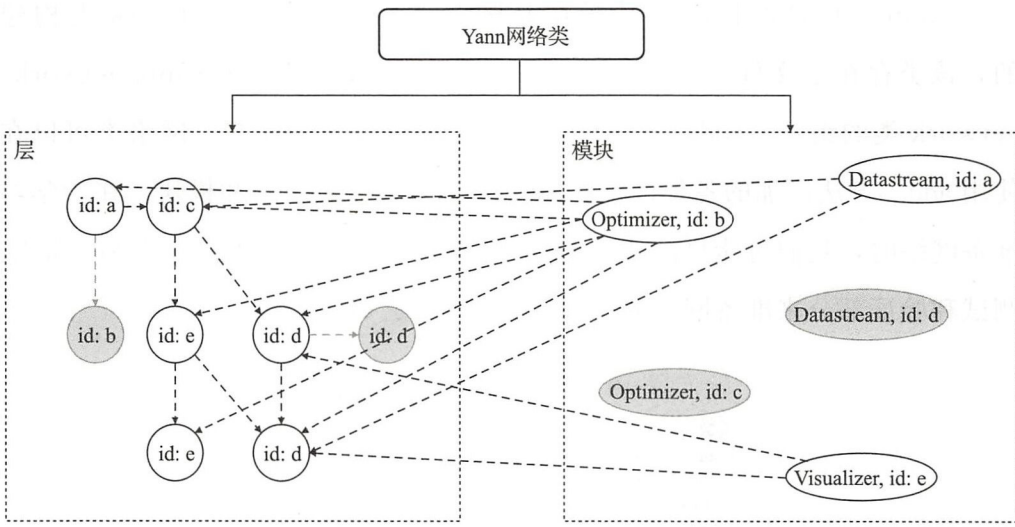


图 A-2　编写的网络。灰色和阴影的对象是网络的未编写部分

快速启动 Yann：logistic 回归

让我们马上看一个例子，与 Yann 建立一个 logistic 回归。Yann 工具箱的开始和结束是 yann.network.network 类。yann.network.network 对象是所有建模发生的地方。从导入 yann.network.network 类并创建一个 yann.network.network 类对象开始：

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用


```
>> from yann.network import network
>> net = network()
```

因此，我们创建了一个新的网络。网络中没有任何层或模块。通过验证 net. layers 属性可以看出这一点。这将产生一个基本上是空字典 {} 的输出。现在让我们添加一些层。工具箱附带了一个 skdata 端口，用于为这个工具箱预先打包的手写字符的 MNIST 数据集。数据集通常位于 _datasets/_dataset_xxxxx。有关如何为 Yann 转换自己的数据集，请参阅在线教程。我们需要添加的第一层是一个输入层。每一个输入层都需要一个数据集与它相关联。让我们创建这个数据集和层，如下代码所示：

```
>> from yann.utils.dataset import cook_mnist
>> cook_mnist()
```

编写数据集。运行此代码将打印一个语句，效果如下：

```
>> Dataset xxxxx is created.
```

在语句中标记 xxxxx 的五个数字是数据集的码字。实际数据集位于现在的数据集 _datasets/_dataset_xxxxx/从该代码被调用的地方。MNIST 数据集被创建并以配置为 Yann 的格式存储在该数据集上。有关如何为 Yann 转换自己的数据集请参见教程。我们需要添加的第一层是一个输入层。每一个输入层都需要一个数据集与它相关联。让我们像这样创建这个层：

```
>> dataset_params= { "dataset": "_datasets/_
  dataset_71367", "n_classes" : 10 }
>> net.add_layer(type = "input", dataset_init_args =
  dataset_params)
```

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

这段代码创建并添加一个新的数据流模块到网络上，并用这个数据流来连接新添加的输入层。让我们添加一个逻辑回归层。Yann 设置的默认分类器是 Logistic 回归分类器。参考工具箱文档或其他类型的教程。现在让我们创建这个分类器层：

```
>> net.add_layer(type = "classifier", num_classes = 10)
>> net.add_layer(type = "objective")
```

层目标从可被用作学习度量的分类器层创建损失函数。它还为其他模块（如优化器模块）提供了一个范围。现在我们的网络被创建和构建，我们可以看到网络对象已经填充了：

```
>> net.layers
>> {'1': <yann.network.layers.classifier_layer object at
  0x7eff9a7d0050>, '0': <yann.network.layers.input_
layer object at 0x7effa410d6d0>, '2': <yann.network.
layers.objective_layer object at 0x7eff9a71b210>}
```

这个字典的键，如“1”“0”和“2”代表 id 的层。我们可以以自定义 id 创建一层，具体做法是为 add_layer 方法提供 id 参数现在已准备好将被训练的网络。训练之前，我们需要建立的优化器和其他工具，我们使用默认的配置。一切准备就绪，在训练前，Yann 要求网络已被“cooked”（编写）：

```
>> net.cook()
```

根据电脑的运行速度，编写可能需要几分钟，并可能打印它正在做的事情。一旦编写，我们可能注意到，例如，网络具有优化器模块：

```
>> net.optimizer
>> {'main': <yann.network.modules.optimizer object at
  0x7eff9a7c1b10>}
```

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

为了训练我们刚刚编写的模型，我们可以使用一旦网络被编写后对我们可用的训练功能：

```
>> net.train()
```

这将为每次迭代打印进程，并且在每次迭代后，显示验证的准确性。该验证集独立于训练集。训练默认运行 40 次：其中 20 次为较高的学习率，另 20 次为微调学习率。

每个层也有一个 layer.output 对象．可以通过使用 layer.activity 方法来探测输出，只要它通过输入层直接或间接与数据流模块相关，并且网络被编写。让我们观察一个试验的输入层的活动。一旦经过训练，我们就能观察到这个输出。层活动将只是一个数字数组，所以让我们打印它的形状：

```
>> net.layer_activity(id = '0').shape
>> net.layers['0'].output_shape
```

第二行代码将验证我们在第一行中产生的输出。一个有趣的层输出是目标层的输出，它将给我们当前网络的负对数似然，我们试图最小化它。

```
net.layer_activity(id = '2')
>> array(2.1561384201049805,dtype=float32)
```

一旦完成训练，我们就可以在测试集上运行网络前馈，以产生泛化的性能结果：

```
>> net.test()
```

我们现在成功地使用了 Yann 工具箱并实现了 logistic 回归。在这里实施的 logistic 回归的完整成熟的代码可以在 pantry/tutorials/log_

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

reg. py 中找到。这段代码在评论中也有其他选项可以提供给我们在这里所做的一些函数调用来更好地解释过程。

多层神经网络

使用 Yann 将 logistic 回归扩展到 MLNN 是一些隐藏层的简单添加。让我们再加上几个。而不是把输入连接到分类器，正如我们在回归示例中看到的那样，让我们添加一些完全连接的隐藏层。隐藏层可以使用层 type= 'dot_product'创建：

```
>> net.add_layer (type = "dot_product",
                  origin ="input",
                  id = "dot_product_1",
                  num_neurons = 800,
                  activation ='relu')

>> net.add_layer (type = "dot_product",
                  origin ="dot_product_1",
                  id = "dot_product_2",
                  num_neurons = 800,
                  activation ='relu')
```

注意所传递的参数。num_neurons 是层中节点的数量。请注意，我们如何使用 id 参数对各层进行模块化。原点表示哪个层将是新层的输入。默认情况下，Yann 假定所有层都是串行输入的，并选择最后添加的层作为输入。使用 origin，可以创建各种类型的体系结构。事实上，任何有向无环图（DAG）都可以手工绘制。现在让我们添加一个分类器和一个目标层：

```
>> net.add_layer (type = "classifier",
                   id = "softmax",
```

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用


```
                origin = "dot_product_2",
                num_classes = 10,
                activation = 'softmax',
                )

>> net.add_layer (type = "objective",
                id = "nll",
                origin = "softmax",
                )
```

我们这次创建自己的优化器模块，而不是使用 Yann 默认值。对于任何在 Yann 中的模块，初始化都可以使用 add_module 方法完成。add_module 方法通常采用一种输入类型，在这种情况下是优化器和一组初始化参数，我们的情况是 params= optimizer_params。任何模块参数，在这种情况下都是 optimizer params，它是相关选项的字典。一个典型的优化器设置是：

```
>> optimizer_params = {
            "momentum_type"       : 'polyak',
            "momentum_params"     : (0.9,0.95,30),
            "regularization"      : (0.0001,0.0002),
            "optimizer_type"      : 'rmsprop',
            "id"                  : 'polyak-rms'
            }

>> net.add_module (type = 'optimizer',params =
  optimizer_params)
```

我们已经成功地加入了具有某些 L_1 和 L_2 范数的 RmsProp 反向传播。一旦添加了优化器，我们就可以编写和学习：

```
>> learning_rates = (0.05,0.01,0.001)
>> net.cook( optimizer = 'main',
          objective_layer = 'nll',
          datastream = 'mnist',
          classifier = 'softmax',
          )
```

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用


```
>> net.train(epochs = (20, 20),
             validate_after_epochs = 2,
             training_accuracy = True,
             learning_rates = learning_rates,
             show_progress = True,
             early_terminate = True)
```

一旦完成，让我们运行 net.test()。这里介绍了一些新的论点，它们大部分是在上下文中容易理解的。epoch 代表一个元组，它是训练的历元数和微调之后的历元数。可能有几个阶段的微调。Yann 用术语“时代”来代表每一个以一个学习速率运行的时代。learning_rates 表示学习速率。这个学习速率的第一个元素是退火参数。学习速率自然有一个长度大于一个时代的长度。show_progress 将打印每个时间段的进度条。在不同的验证数据集上的许多不同的时期之后，验证周期将执行验证。本教程的完整代码有附加注释，可以在文件 pantry.tutorials.mlp.py 中找到。运行代码如下：

```
>> from pantry.examples.mlp import mlp
>> mlp(dataset = 'some dataset created')
```

卷积神经网络

既然我们已经完成了基础，把这个扩展到 LeNet-5 型网络并不难。我们需要做的是添加几个对流层，并且可以使用相同的 add_layer 方法来添加：

```
>> net.add_layer (type = "conv_pool",
                  origin = "input",
                  id = "conv_pool_1",
                  num_neurons = 20,
                  filter_size = (5,5),
```

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用


```
                    pool_size = (2,2),
                     activation = 'tanh',
                     verbose = verbose
                     )

>> net.add_layer ( type = "conv_pool",
                     origin = "conv_pool_1",
                     id = "conv_pool_2",
                     num_neurons = 50,
                     filter_size = (3,3),
                     pool_size = (2,2),
                     activation = 'tanh',
                     verbose = verbose
                     )
```

现在可以将 conv_pool_2 添加到上面讨论的 MLNN 体系结构中，我们有一个 CNN。CNN 将在 MNIST 数据集上产生超过 99%的泛化精度。

工具箱有更多的选项，这里没有讨论，使用它们的描述可以在 Yann 在线文档中找到。同样的代码在教程的目录中给出。

自动编码器

自动编码器比其他网络要复杂一些。初始步骤是相同的，直到第一层（或多个层）的自动编码器。解码器层现在需要把编码器层的同一组参数作为输入，并在其中进行转置。下面的代码块使用 Yann 创建这样的层：

```
>> net.add_layer ( type = "dot_product",
                   origin = "encoder",
                   id = "decoder",
                   num_neurons = 784,
                   activation = 'tanh',
                   input_params = [net.
                     layers['encoder'].w.T,None],
                   # Use the same weights but
                     transposed for decoder.
                   learnable = False,
                   verbose = verbose
                   )
```

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

因此，我们增加了一个解码层，它将图像解码成 784 个输出，并作为输入的转置权重的输入。没有一个对应于偏置。如果需要，我们也可以强制编码器偏置。

为创造一个损失函数，我们可以重塑或压缩图像使之恢复到正方形。通过以下代码利用 unflatten layer 完成这一操作。

```
>> net.add_layer ( type = "unflatten",
                  origin = "decoder",
                  id = "unflatten",
                  shape = (28,28,1),
                  verbose = verbose
                  )
```

若不平坦，图像需要与输入进行比较，这应该是学习的目的。我们可以利用目标层中的合并层和 layer_type 参数来完成这项任务：

```
>> net.add_layer ( type = "merge",
                  origin = ("input","unflatten"),
                  id = "merge",
                  layer_type = "error",
                  error = "rmse",
                  verbose = verbose)

>> net.add_layer ( type = "objective",
                  origin = "merge",
                  id = "obj",
                  objective = None,
                  layer_type = 'generator',
                  verbose = verbose
                  )
```

自动编码器就这样创建出来了。在编写和训练后，如果再使用一个可视化工具，我们就可以可视化保存和生成的图像。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

总结

上面的示例旨在快速演示这里描述的一些网络及其类型。这些代码的完整代码可在代码库的教程目录中找到。工具箱比这里描述得详细，并且有更多的选项；然而，这里只介绍了选定的项目。请参考 http://www.yann.network 的代码文档，了解更多的细节和其他前沿开发教程和代码。

参考文献

Abadi, Martin, Agarwal, Ashish, Barham, Paul et al. 2016. *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.* arXiv preprint arXiv:1603.04467.

Bergstra, James, Breuleux, Olivier, Bastien, Frederic et al. 2010. Theano: A CPU and GPU math compiler in Python. *Proceedings of the 9th Python in Science Conference*, pp. 1–7.

Chollet, François. 2015. *Keras.* GitHub repository. https://keras.io/.

Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff et al. 2014. Caffe: Convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM International Conference on Multimedia.* ACM, pp. 675–678.

Lasagne. n.d. *Lasagne.* http://lasagne.readthedocs.org/en/latest/.

Vedaldi, Andrea and Lenc, Karel. 2015. MatConvNet: Convolutional neural networks for MATLAB. *Proceedings of the 23rd ACM International Conference on Multimedia.* ACM, pp. 689–692.

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

后　记

在机器学习领域，存在着一些重要的认识哲学。在本书中，我们已经研究探讨了连接机制哲学及其对计算机视觉的应用。第 1 章介绍了不同形式的图像表示，以及一些非连接图像模型。第 2 章探讨了线性回归、优化和正则化。第 3 章介绍了感知器或计算神经元、多层感知器或多层神经网络，以及它们通过反向传播算法学习的方法。当然还包括一些实用技巧，这些技巧使学习变得更好、更快、更稳定。第 4 章介绍了卷积池化层和卷积神经网络（CNN），并研究了一些相关经典案例。第 5 章主要展示了一些预训练网络的新颖使用案例，这些网络是在实践中部署 CNN 最常见的方式之一。第 5 章还介绍了基于 CNN 的生成模型。

人工神经网络是机器学习的经典领域之一。过去的几十年里，它的普及度和使用频率却是起起伏伏。人工神经网路经常被错误地宣传和炒作，而它真正的意义却被我们忽略。这些错误的认识，大部分都是因为人们没有认真地研究过它。在人工神经网络这几十年的发展中，也产生了一些关于点对点神经网络应用于计算机视觉的争论。其中，最激烈的争论可能就是最近的 LeCun（2015）指出的“人工神经网络一直在学习”。如果读者对连接主义的历史感兴趣，推荐阅读《Talking Nets：An Oral History of

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Neural Networks》（Anderson 和 Rosenfeld，2000）。

关于连接主义及其对人类认知的影响，一直存在着哲学上的难题。Fodor 和 Pylyshyn（1998）的一篇文章对其中的一些问题提出了一个非常好的（虽然颇具争议）参考。尽管这篇文章有一些很有争议的批评，它的一些重要前提也已经被证明是错误的，但它确实给那些为神经网络寻求哲学基础的人提供了很好的见解。在本书中，虽然我们有意地回避这些哲学，把更多的注意力放在神经网络工程上，但作者还是会探讨哲学经典辩论在现代神经网络上的体现。

第 5 章介绍了对预训练网络进行微调的思路，而不是从头开始初始化一个新网络。这引发了一些对盲目使用预训练网络来进行计算机视觉任务的争论。通常情况下，只有通过实现经验（即所谓的“脏活累活”），工程师才能对何时使用预先培训网络产生一种直觉。是使用预训练的网络，还是针对每项任务从头开始训练一个网络，在这场争论中，对于大多数计算机视觉任务来说，就如第 5 章所探讨的，似乎从某种程度上都应该选择预训练的网络。基于这一点，在尝试从头开始构建和培训自己的架构之前，总会强烈建议工程师先对一个相关的网络进行微调。

在关于使用神经网络来研究人类认知的争论中，也出现了一个经常被提及的问题，就是神经网络概括潜在数据的能力。这与系统性的争论是对立的，它的支持者认为神经网络仅仅是建立联系，而不会学习结构或句法，参见 Fodor 和 Pylyshn（1988）的论文。所以一些早期的论据被拿出来用以支持论证神经网络（Niklasson 和 Gelder，1994）。神经网络已经证明了其强大的归纳概括能力，预训练网络和生成器网络从复杂分布中生成

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

样本的能力的最好例子是生成对抗网络（GAN）（Goodfellow 等人，2014）。在这个广泛变化的场景中，我们需要新颖的方法去思考概括归纳。大多数开发神经网络的工程师，通常都没有足够的数据来训练相关的大型网络。所以使用预训练网络会是一个不错的解决方案。另外一个比较使用的方法就是在训练时从相关领域中获取数据。如果可以很容易地获得该训练数据，以便学习更好的表示，然后将其应用于测试数据的领域，那么这可能会提供另一种解决问题的方法。领域适应和转移学习理念的研究已经成为一个非常有潜力和让人兴奋的领域。

神经网络另一个令人兴奋的研究和发展领域便是强化学习。自从 DeepMind 等公司的科技展示显示出神经网络已经可以进行游戏的学习，甚至可以在游戏中比人类玩家发挥得更好，强化学习就慢慢变得流行起来。让神经网络学习玩游戏并不是一个新概念，从一开始它就一直是一个活跃的领域，从 Arthur Samuel 的跳棋到谷歌基于 DeepMind 的 AlphaGo，神经网络在几场游戏中都被证明是成功的。数据是庞大丰富的。在强化学习中，我们可以模拟一个比实际时间更快的世界。研究界一直期待着一个成熟的系统，它能从这些领域中学习，同时确保在这里学到的东西可以应用到现实世界中。从合成数据中学习，也激发了对 GAN 的使用，以创建预先训练的网络。合成数据可以让网络在拥有非常少量数据的情况下进行学习，这也是目前被研究的热点领域。另一个热门领域（可能是神经网络研究的下一个前沿）便是无监督学习。随着移动电话和图像共享社交媒体的兴起，产生的图像内容数量呈指数级的增长。为如此巨大的原始数据贴上标签是一项非常艰巨的任务，因此研究人员才着手无监督学习的研究，以挖掘这些数据的潜力。对于感兴趣的人来说，这将

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

是另一个令人兴奋的领域。

最近大家对神经网络的兴趣激增，除了神经网络技术本身之外，还存在其他技术外的几个因素。其中值得一提的也是我们非常欣赏的一点是研究界对知识材料分享统一性的意识。来自学术界和工业界的开源软件文化，极大地提高了讨论灵活性和讨论速度，以及结果的再现性。这种文化（加上 GitHub、travis-ci 和 readthedocs 等工具的可靠代码共享）已经对学界产生了积极的影响，并在很大程度上更好地促进了学术协作和传播。因此，我们鼓励读者追随并延续这种文化。

虽然神经网络在一些任务的端到端学习中被证明是可靠的，但有些任务仍然需要手工操作。并不是所有的系统都可以完全从数据中进行端到端的学习。而“深度学习可以做到人类在 1 秒中内能做的任何事情”（Ng，2016），这似乎也暗示，对于目前的深度学习来说，如果一个任务涉及太多的人类很长时间才能完成的认知和逻辑思维，那么这项任务可能就没那么轻松容易了。虽然这个结论是非正式的，但我们认为，这可能仍然是当前深度学习系统的一个良好的基准。尽管我们可能无法预言深度学习和神经网络对未来的影响，但能够做到人类 1 秒钟内所能做的事情，也算是一个不错开始。

参考文献

Anderson, James A and Rosenfeld, Edward. 2000. *Talking Nets: An Oral History of Neural networks.* Cambridge, MA: MIT Press.

Chalmers, David. 1990. Why Fodor and Pylyshyn were wrong: The simplest refutation. *Proceedings of the Twelfth Annual Conference of the Cognitive*

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Science Society, pp. 340–347.

Fodor, Jerry A and Pylyshyn, Zenon W. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition* (Elsevier) 28: 3–71.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi et al. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems,* pp. 2672–2680.

LeCun, Yann. 2015. It's learning all the way down. *International Conference on Computer Vision.*

Ng, Andrew. 2016. Nuts and bolts of deep learning. *NIPS 2016, Deep/Bay Area Deep Learning School.*

Niklasson, Lars F and Gelder, Tim. 1994. On being systematically connectionist. *Mind & Language* 9: 288–302.

仅供非商业用途或交流学习使用

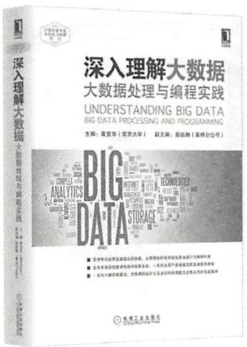
仅供非商业用途或交流学习使用

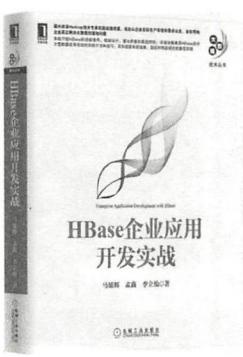
推荐阅读

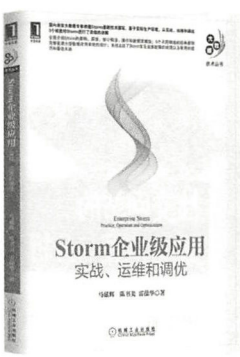
仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

推荐阅读







仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

译者简介

钱亚冠

博士、副教授、硕士生导师、中国计算机学会会员、中国自动化学会会员。毕业于浙江大学计算机系，获工学博士学位。主要研究方向为人工智能安全、机器学习与大数据处理。已发表 SCI/EI 论文 20 余篇。参与国家 973、863、国家自然科学基金项目 5 项。主持浙江省自然科学基金、浙江省教育厅科研项目、浙江省云计算工程技术中心开放课题多项，相关成果获 IEEE 第三届智慧城市国际会议优秀成果奖。

王滨

博士 / 博士后、教授级高工、海康威视副总裁、国家重大专项“网络空间安全”评审专家。主要研究方向人工智能安全、物联网安全等。

仅供非商业用途或交流学习使用

仅供非商业用途或交流学习使用

Convolutional Neural Networks
in Visual Computing

本书覆盖内容主要包括设计和开发卷积神经网络架构的一些基本知识。对于那些想要学习或构建深度视觉系统的工程师或学生来说，这本书会是一本非常好的入门读物，可以帮助他们快速地进入这个领域。本书还提供了丰富的理论知识和实操案例，以及一系列完备的工具包，以帮助初学者获得在理解和构建卷积神经网络（CNN）时所必要的基本信息。本书的重点将集中在卷积神经网络的基础部分，而不会涉及在高级课程中才出现的一些概念（CNN 相关话题）。

本书特点：

- 为学习和开发 CNN 提供足够基础和全面的知识。
- 为非计算机科学专业的人提供成体系和系统的知识内容。
- 易于理解，通过案例研究、代码片段及数据集进行生动的展示和说明。
- 适合于那些想要在该领域锻炼自己实操能力的高年级学生或刚从业的人员。
- 本书包含 CNN 概念、方法和软件工具集，可以帮助你学习、理解和实现前沿的 CNN 技术。

CRC Press
Taylor & Francis Group

投稿热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn

上架指导：计算机/人工智能
ISBN 978-7-111-61239-1
9 787111 612391
定价：59.00元

仅供非商业用途或交流学习使用